

Vinícius Ávila da Silva

Denoising Unidimensional por Esparsificação no Domínio Wavelet

Brasil

Vitória, 2018

Vinícius Ávila da Silva

Denoising Unidimensional por Esparsificação no Domínio Wavelet

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Universidade Federal do Espírito Santo – UFES

Programa de Pós Graduação em Engenharia Elétrica – PPGEE

Laboratório de Computação e Sistemas Neurais – CISNE

Orientador: Evandro Ottoni Teatini Salles

Brasil

Vitória, 2018

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

S586d Silva, Vinícius Ávila da, 1992-
Denoising unidimensional por esparsificação no domínio
wavelet / Vinícius Ávila da Silva. – 2018.
64 f. : il.

Orientador: Evandro Ottoni Teatini Salles
Dissertação (Mestrado em Engenharia Elétrica) –
Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Algoritmos. 2. Wavelets (Matemática). 3. Matrizes
esparsas. 4. Estatística não paramétrica. 5. Estimador não-
viesado do risco de Stein. I. Salles, Evandro Ottoni Teatini.
II. Universidade Federal do Espírito Santo. Centro Tecnológico.
III. Título.

CDU: 621.3

Vinícius Ávila da Silva

Denoising Unidimensional por Esparsificação no Domínio Wavelet

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Prof. Dr. Evandro Ottoni Teatini
Salles
UFES
Orientador

Prof. Dr. Rodrigo Varejão Andreão
IFES

Prof. Dr. Luiz Alberto Pinto
IFES

Brasil
Vitória, 2018

Agradecimentos

Primeiramente gostaria de agradecer à minha esposa, Larisse, pelo apoio e paciência ao longo desses anos, por estar sempre ao meu lado e servir a Deus junto comigo.

À toda a minha família, especialmente aos meus pais, Vitor e Angela, pelo suporte emocional ilimitado que sempre mostraram. Sem eles, eu certamente não teria chegado até aqui.

Ao meu orientador, Dr Evandro, pela paciência e atenção que demonstrou durante o meu curso. Passarei o restante da minha vida tentando emular seu nível de conhecimento e tranquilidade.

Ao CNPq pela bolsa de estudos sem a qual não seria possível minha permanência do programa.

Finalmente, agradeço a Deus pelo dom da vida e do conhecimento, pela força e esperança em todos os momentos desta vida.

*"O Senhor controla todo evento aleatório, seja ele fruto deliberado
de uma ação humana, como um lançar de dados ou moedas,
seja apenas uma coincidência, como um fio de cabelo
saindo da cabeça de alguém e caindo no chão."*

— Vern Poythress.

Resumo

Um algoritmo de *denoising* busca a remoção ou atenuação do ruído em sinais, sendo utilizado especialmente para o ruído branco. Para sinais unidimensionais, a transformada discreta de *wavelet* (DWT) e a transformada de Fourier de tempo curto (STFT) são as principais transformações utilizadas no *denoising*, e ambas apresentam diversos parâmetros que devem ser definidos pelo usuário. Devido à grande influência que estes parâmetros exercem sobre o desempenho do algoritmo, propõe-se neste trabalho o desenvolvimento de uma variação do *denoising* por DWT na qual os parâmetros de base e escala são adaptados de forma a maximizar a esparsidade da representação do sinal no domínio *wavelet*. Devido a ortogonalidade da transformação, a norma l_1 foi utilizada como medida objetiva de esparsidade. Duas variações do *denoiser* foram apresentadas, em função do número de bases que compõem o dicionário, e testes com sinais diversos foram realizados para uma comparação com o *denoising* por bloco tempo-frequência em termos de desempenho e custo computacional. Os resultados obtidos mostraram que as técnicas propostas apresentaram desempenho, em média, maior que o *denoising* por bloco tempo-frequência. Com o auxílio de teste estatístico não-paramétrico de Wilcoxon, concluiu-se que o uso de um dicionário reduzido não afeta significativamente o desempenho, mesmo com a redução no tempo de processamento de aproximadamente quatro vezes.

Palavras-chave: Denoising. Wavelet Shrinkage. Estimador Não-polarizado de Risco de Stein. Esparsidade.

Abstract

A denoising algorithm seeks to remove or reduce noise from signals, and it's specially used for white noise. For one-dimensional signals, the discrete wavelet transform (DWT) and the short-time Fourier transform (STFT) are the main transformations used in denoising, and both present several parameters that should be selected by the user. Due to the great influence those parameters have on the algorithm's performance, the proposal of this work is to develop a variation of the DWT denoising (wavelet shrinkage) in which the basis and scale parameters are adapted to maximize the sparsity of the signal's representation in the wavelet domain. Due to the orthogonality of the transformation, the l_1 norm was used as an objective sparsity measure. Two variations of the denoiser were presented, with respect to the number of basis that make up the dictionary. Tests were performed on several signals for a comparison with the time-frequency block denoising in terms of performance and computational cost. The results showed that the proposed techniques presented, on average, higher mean performance than the time-frequency block denoising. With the use of the Wilcoxon non-parametric statistical test, it was concluded that the use of a reduced dictionary does not significantly affect the performance, even with the reduction of the processing time by four times, approximately.

Keywords: Denoising. Wavelet Shrinkage. Stein Unbiased Estimate of Risk. Sparsity.

Lista de ilustrações

Figura 1 – Fluxograma generalizado de algoritmos de <i>denoising</i>	13
Figura 2 – Sinal <i>blocks</i> original e com ruído.	17
Figura 3 – Diagrama de blocos da transformada <i>wavelet</i> discreta.	18
Figura 4 – Sinal <i>blocks</i> original e com ruído decomposto na função Haar.	19
Figura 5 – Diagrama de blocos da transformada <i>wavelet</i> discreta inversa.	20
Figura 6 – Diagrama de blocos da transformada <i>wavelet</i> discreta na escala 3.	21
Figura 7 – Sinal <i>blocks</i> original e com ruído decomposto na função Haar em escala 3.	22
Figura 8 – Funções Daubechies para $1 \leq p \leq 10$	23
Figura 9 – Funções Symlets para $2 \leq p \leq 10$	23
Figura 10 – Funções Coiflet para $1 \leq p \leq 5$	24
Figura 11 – Função discreta de Meyer.	24
Figura 12 – Funções de ativação de limiarização.	26
Figura 13 – Sinal <i>blocks</i> decomposto na função Haar após limiarização.	27
Figura 14 – Sinal <i>blocks</i> recuperado por função Haar e SURE.	27
Figura 15 – Sinal <i>blocks</i> recuperado por função Haar em escala 5.	28
Figura 16 – Sinal <i>blocks</i> recuperado por função Haar em escala 10.	29
Figura 17 – Funções <i>wavelet</i> db1 e db4.	30
Figura 18 – Sinal <i>blocks</i> decomposto nas funções db1 e db4.	31
Figura 19 – Sinais <i>blocks</i> , <i>bumps</i> , <i>doppler</i> e <i>heavy sine</i> originais e com ruído.	34
Figura 20 – Desempenho do <i>wavelet shrinkage</i> em função da esparsidade para os sinais <i>blocks</i> , <i>bumps</i> , <i>doppler</i> e <i>heavy sine</i>	35
Figura 21 – Sinais <i>blocks</i> , <i>bumps</i> , <i>doppler</i> e <i>heavy sine</i> após <i>denoising</i>	36
Figura 22 – Metodologia para definição de dicionário conciso.	38
Figura 23 – Eletrocardiograma gerado pelo ECGSYN/PhysioNet.	39
Figura 24 – Desempenho médio para o sinal <i>blocks</i>	40
Figura 25 – Desempenho médio para o sinal <i>bumps</i>	41
Figura 26 – Desempenho médio para o sinal <i>doppler</i>	42
Figura 27 – Desempenho médio para o sinal <i>heavy sine</i>	43
Figura 28 – Desempenho médio para o sinal ECG.	44
Figura 29 – Desempenho médio para o sinal ECG a 1024Hz.	45
Figura 30 – Desempenho médio para o sinal Mozart.	46
Figura 31 – Desempenho médio para o sinal Mozart com desvio padrão real do ruído e escala máxima de 2.	47
Figura 32 – Tempo de processamento médio para o sinal <i>blocks</i>	48
Figura 33 – Tempo de processamento médio para o sinal <i>bumps</i>	49
Figura 34 – Tempo de processamento médio para o sinal <i>doppler</i>	50

Figura 35 – Tempo de processamento médio para o sinal <i>heavy sine</i>	51
Figura 36 – Tempo de processamento médio para o sinal ECG.	52
Figura 37 – Tempo de processamento médio para o sinal ECG a 1024Hz.	53
Figura 38 – Tempo de processamento médio para o sinal Mozart.	54
Figura 39 – Tempo de processamento médio para o sinal Mozart com desvio padrão real do ruído e escala máxima de 2.	55

Lista de tabelas

Tabela 1 – SNR (dB) do sinal <i>blocks</i> recuperado por 6 bases em 5 níveis distintos. O melhor e o pior resultados estão destacados em negrito e itálico, respectivamente.	28
Tabela 2 – SNR (dB) do sinal <i>blocks</i> recuperado pela função Haar em 10 níveis distintos.	29
Tabela 3 – Critérios desejáveis de medidas de esparsidade.	32
Tabela 4 – Sinais para teste.	38
Tabela 5 – SNR média (dB) do sinal <i>blocks</i>	41
Tabela 6 – SNR média (dB) do sinal <i>bumps</i>	41
Tabela 7 – SNR média (dB) do sinal <i>doppler</i>	42
Tabela 8 – SNR média (dB) do sinal <i>heavy sine</i>	43
Tabela 9 – SNR média (dB) do sinal ECG.	44
Tabela 10 – SNR média (dB) do sinal ECG a 1024Hz.	45
Tabela 11 – SNR média (dB) do sinal Mozart.	46
Tabela 12 – SNR média (dB) do sinal Mozart com desvio padrão real do ruído e escala máxima de 2.	47
Tabela 13 – Tempo de processamento médio (ms) do sinal <i>blocks</i>	48
Tabela 14 – Tempo de processamento médio (ms) do sinal <i>bumps</i>	49
Tabela 15 – Tempo de processamento médio (ms) do sinal <i>doppler</i>	50
Tabela 16 – Tempo de processamento médio (ms) do sinal <i>heavy sine</i>	51
Tabela 17 – Tempo de processamento médio (ms) do sinal ECG.	52
Tabela 18 – Tempo de processamento médio (ms) do sinal ECG a 1024Hz.	53
Tabela 19 – Tempo de processamento médio (ms) do sinal Mozart.	54
Tabela 20 – Tempo de processamento médio (ms) do sinal Mozart com desvio padrão real do ruído e escala máxima de 2.	55

Lista de abreviaturas e siglas

DWT	<i>Discrete Wavelet Transform</i>
FFT	<i>Fast Fourier Transform</i>
IDWT	<i>Inverse Discrete Wavelet Transform</i>
ML	<i>Maximum Likelihood</i>
RMSE	<i>Root Mean Square Error</i>
SNR	<i>Signal-to-Noise Ratio</i>
STFT	<i>Short-Time Fourier Transform</i>
SURE	<i>Stein Unbiased Estimate of Risk</i>

Sumário

1	INTRODUÇÃO	13
1.1	Motivação	15
1.2	Hipótese e Objetivos	15
1.3	Estrutura da Dissertação	15
2	WAVELET SHRINKAGE	17
2.1	Transformada <i>Wavelet</i> Discreta	17
2.1.1	Forma Matricial	20
2.1.2	Análise Multiresolução	21
2.1.3	Bases Ortogonais	22
2.2	Limiarização	24
2.2.1	Estimador Não-enviesado de Risco de Stein	25
2.2.2	Aplicação de Limiares	26
2.3	Influência dos Parâmetros	28
3	MATERIAIS E MÉTODOS	30
3.1	Esparsidade no Domínio <i>Wavelet</i>	30
3.1.1	Medidas de Esparsidade	31
3.1.2	Esparsidade e Desempenho	33
3.1.3	Minimização da Norma L1	37
3.2	Delineamento Experimental	38
4	RESULTADOS E DISCUSSÕES	40
4.1	Relação Sinal-Ruído	40
4.2	Tempo de Processamento	48
5	CONCLUSÕES E TRABALHOS FUTUROS	56
5.1	Trabalhos Futuros	57
	REFERÊNCIAS	58
	APÊNDICE A – PSEUDO-CÓDIGOS	61
	APÊNDICE B – TESTE ESTATÍSTICO DE WILCOXON	64

1 Introdução

O problema de sinais contaminados por ruído é comum em diversas aplicações. Distorções inevitáveis ou de difícil atenuação podem ser causadas pela presença de ruído de fundo ou pelos próprios equipamentos de captura e processamento de sinais. Estas perturbações não apenas diminuem a qualidade de sinais como músicas e fotografias, mas potencialmente reduzem a eficiência de sistemas de reconhecimento de voz, telemedicina, transmissão de dados, entre outros.

Tradicionalmente, duas técnicas de processamento de sinais são utilizadas para o tratamento de sinais ruidosos: a filtragem e a limiarização no domínio do tempo (*noise gate*). Um filtro clássico atenua ou amplifica componentes em função de sua frequência, sem distinção entre o sinal desejado e o ruído. Assim, a filtragem normalmente utilizada (passa-baixas) é eficiente na atenuação das componentes rápidas do ruído, porém afeta também o sinal original em suas altas frequências. O *noise gate* atenua as componentes temporais de um sinal que possuem amplitude inferior a um limiar determinado. Portanto, mitiga o ruído apenas em instantes nos quais a amplitude do sinal ruidoso é inferior ao limiar.

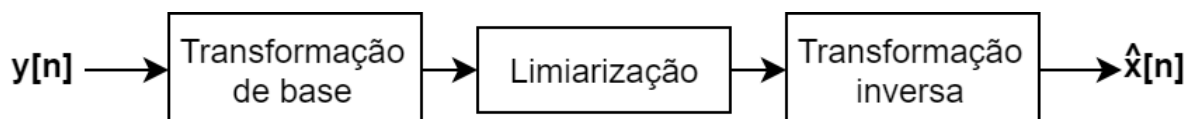
Como o ruído branco (amostras descorrelatadas) possui componentes em todos os instantes e densidade espectral de potência uniforme, não é eficientemente tratado pelas duas técnicas clássicas. Devido a importância e recorrência deste tipo de ruído, se fez necessário o desenvolvimento dos algoritmos de *denoising*, que buscam remover ruído, erros e perturbações de um sinal, segundo Metzler, Maleki e Baraniuk (2016). Semelhantemente, se a observação $y[n]$ de um sinal $x[n]$ contaminado por um ruído aditivo $e[n]$ é dada por

$$y[n] = x[n] + e[n], \quad (1.1)$$

o objetivo de um algoritmo de *denoising* é estimar um vetor $\hat{x}[n]$ com erro, em relação ao sinal $x[n]$, inferior à condição anterior.

As técnicas modernas de *denoising* são variações de um modelo genérico composto por três etapas: transformação de base, limiarização e transformação inversa, conforme Figura 1.

Figura 1 – Fluxograma generalizado de algoritmos de *denoising*.



Fonte: Adaptado de Krim et al. (1999).

Esta metodologia originou-se com os trabalhos de Donoho e Johnstone que, segundo Mallat (2008), provaram que a limiarização de representações esparsas pode gerar estimadores quase ótimos. A etapa de transformação, portanto, tem como objetivo levar o sinal a um domínio cuja representação seja esparsa e, por consequência, mais significativa que o ruído branco. Se isto for alcançado, um limiar propriamente definido pode melhorar a relação sinal-ruído (SNR) do sinal, definida por

$$SNR = 10 \log_{10} \left(\frac{VAR(\hat{x}[n])}{VAR(\hat{x}[n] - x[n])} \right). \quad (1.2)$$

Dentre as etapas de um *denoiser* genérico (Figura 1), a síntese do sinal consiste apenas na aplicação da transformação já utilizada na primeira etapa, porém de forma inversa. Assim, o desenvolvimento de um algoritmo de *denoising* consiste na definição dos dois primeiros estágios: transformação e limiarização.

Quanto à primeira etapa, a transformação pode partir de bases pré-definidas ou de um dicionário super completo que é adaptado em função de um conjunto de sinais de treinamento. Enquanto esta opção é eficiente na obtenção de representações esparsas, o custo computacional é elevado. O uso de funções base pré-definidas é mais simples e rápido, em especial na etapa de transformação inversa (AHARON et al., 2006).

Com o uso de transformações com bases conhecidas, Donoho e Johnstone (1994) propuseram a utilização da transformada *wavelet*, o que deu início ao refinamento de estimadores não-lineares (MALLAT, 2008). Para imagens, a transformada *wavelet* também pode ser utilizada (CHANG; YU; VETTERLI, 2000), assim como a codificação esparsa (HYVÄRINEN; HOYER; OJA, 1999), decomposição por valor singular (HE et al., 2011), transformada *curvelet* (STARCK; CANDÈS; DONOHO, 2002), entre outras. Em sinais unidimensionais, o número de transformações é reduzido, de forma que as técnicas desenvolvidas constituem-se de variações do método de Donoho e Johnstone (1995), que utiliza a transformada *wavelet*. Em 2008, porém, Yu, Mallat e Bacry (2008) propuseram a utilização da transformada por bloco tempo-frequência, baseada na transformada de Fourier de tempo curto (STFT), para *denoising* de áudio. Neste trabalho, destaca-se a aplicação da técnica em um sinal de gravação de oboé com SNR inicial de $-2,73\text{dB}$ que, após o processamento, resultou em SNR $11,9\text{dB}$.

Na etapa de limiarização, técnicas como o estimador de risco de Stein (DONOHO; JOHNSTONE, 1995), estimação por máxima verossimilhança (ML) (HYVÄRINEN; HOYER; OJA, 1999) e Minimax (SARDY, 2000) podem ser aplicadas, no entanto o estimador não-enviesado de risco de Stein (SURE) é a técnica mais amplamente utilizada (ZHANG; DESAI, 1998; LUISIER; BLU; UNSER, 2007; BIGOT; DELEDALLE; FÉRAL, 2017). Segundo Metzler, Maleki e Baraniuk (2016), o SURE mostrou ser um método poderoso para definir os parâmetros de um *denoiser*.

1.1 Motivação

Apesar da existência de diversas técnicas de *denoising*, dois problemas são frequentemente encontrados: falta de adaptabilidade e especificidade dos sinais a serem tratados. O *denoising* por transformada *wavelet*, por exemplo, apresenta simplicidade de implementação, porém depende grandemente de parâmetros a serem definidos pelo usuário (Capítulo 2). O *denoising* por bloco tempo-frequencial, por sua vez, tem desempenho muito dependente de uma boa estimativa inicial da potência do ruído, além de não apresentar a mesma eficiência para sinais diversos (Capítulo 4).

Portanto, se faz necessário o desenvolvimento de uma técnica robusta que se adapte às características de sinais diversos, em especial quando o conhecimento do sinal por parte do usuário é pequeno.

1.2 Hipótese e Objetivos

Como o estimador de risco de Stein (SURE) é adaptativo e não apresenta restrições quanto ao tipo de sinal, a hipótese deste trabalho é de que a adaptabilidade e a diversidade de um algoritmo de *denoising* são resultados da etapa de transformação.

Em vista disso, o objetivo do presente trabalho é desenvolver um método de *denoising* de sinais unidimensionais que adeque a base de transformação e seus parâmetros às características do sinal de forma que resultados satisfatórios sejam alcançados em uma grande variedade de sinais. A transformada *wavelet* discreta (DWT), devido ao grande número de bases disponíveis e sua capacidade de decompor sinais com características multiescalares, representa bem diversos sinais. Além disto, apresenta algoritmo de decomposição rápido com custo computacional inferior ao algoritmo rápido da transformada de Fourier (FFT). Por esses motivos, a transformada *wavelet* foi escolhida para o método proposto. Para a posterior adaptação da DWT, um objetivo específico é avaliar a relação entre a esparsidade e o desempenho do *denoising* por DWT.

1.3 Estrutura da Dissertação

Este trabalho organiza-se da seguinte forma. No Capítulo 2, o método de *denoising* por DWT é apresentado. Exemplos de aplicação são expostos com a finalidade de demonstrar a eficiência da técnica e a influência dos parâmetros de transformação. A limiarização por SURE também é apresentada neste capítulo. No Capítulo 3 é analisada a relação entre a esparsidade dos coeficientes *wavelet* e o desempenho do *denoiser*, assim como a metodologia de adaptação da base de transformação, fruto dessa análise. Os sinais gerados para teste e a descrição dos experimentos realizados são expostos, também, neste capítulo. O Capítulo 4 exhibe os resultados do método proposto em comparação com o método por

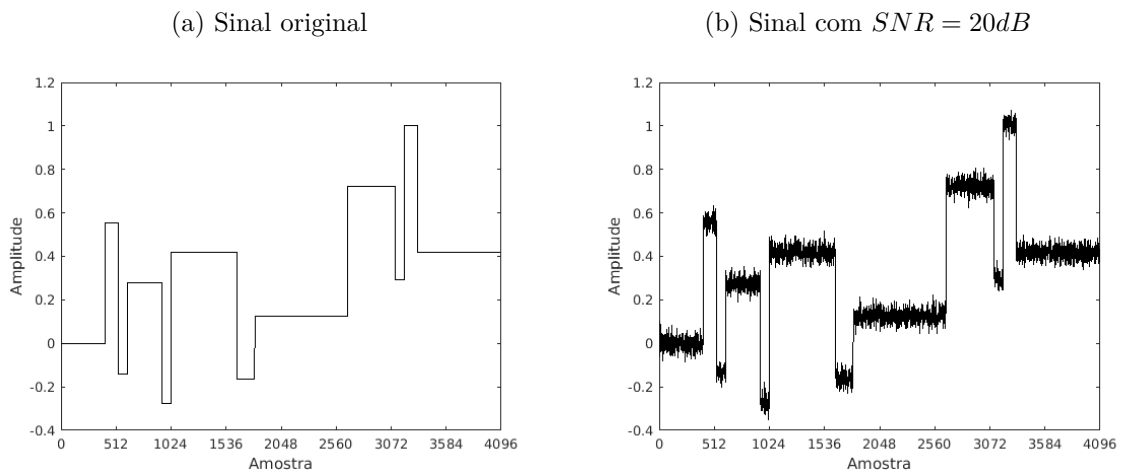
bloco tempo-frequencial de Yu, Mallat e Bacry (2008). Por fim, as conclusões e lacunas para trabalhos futuros são expostas no Capítulo 5.

2 Wavelet Shrinkage

O *denoising* por transformada *wavelet* foi desenvolvido por Donoho e Johnstone (1994) e em seus trabalhos posteriores e consiste na aplicação da DWT no sinal para que poucos coeficientes possuam, em relação aos demais, alta energia. Assim, os componentes com amplitudes altas apresentarão grande SNR e devem ser preservados, enquanto os componentes com baixas amplitudes tem SNR pequena e serão desconsiderados. Devido a redução dos coeficientes de baixa energia, o método foi denominado *wavelet shrinkage*. Após a eliminação de componentes pouco significativos (limiarização), a transformada *wavelet* discreta inversa (IDWT) é utilizada para levar o sinal ao domínio temporal. Portanto, o *wavelet shrinkage* apresenta estrutura conforme o modelo da Figura 1.

Para a demonstração das etapas do método, será utilizado um sinal *blocks* definido por Donoho e Johnstone (1995) com 2^{12} amostras e $SNR = 20dB$ (Figura 2).

Figura 2 – Sinal *blocks* original e com ruído.



Fonte: O próprio autor

A amplitude dos sinais expostos foi normalizada em função do valor de pico do sinal original (Figura 2a).

2.1 Transformada *Wavelet* Discreta

A transformada *wavelet* é uma ferramenta que nasceu da necessidade de uma análise tempo-frequencial multi escalar de sinais. Enquanto a STFT apresenta a localização temporal de componentes de frequência com o uso de uma janela de duração constante, a transformada *wavelet* apresenta janelas temporais variantes com a frequência. Esta propriedade permite que sinais com ampla faixa de frequência sejam analisados com maior

eficiência que a STFT, pois a análise de faixas diferentes de frequência dependem de diferentes tamanhos de janelas (DAUBECHIES, 1992).

Matematicamente, a transformada *wavelet* de uma função de tempo contínuo f com energia finita consiste na sua decomposição em dilatações e translações de uma função base ψ , chamada de *wavelet* mãe, que deve apresentar energia finita e média zero. Com os fatores de dilatação e translação sendo s e u , respectivamente, o dicionário de funções base é dado por (MALLAT, 2008):

$$\mathcal{D} = \left\{ \psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) \right\}_{u \in \mathbb{R}, s \in \mathbb{R}^+}. \quad (2.1)$$

Definida a função ψ , a transformada contínua de *wavelet* da função f é dada por:

$$Wf(u, s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-u}{s}\right) dt, \quad (2.2)$$

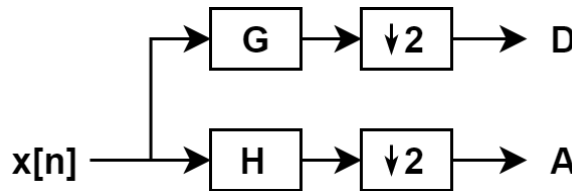
onde ψ^* consiste no conjugado complexo de ψ .

Em sinais discretos, a transformada *wavelet* equivalente a Equação 2.2 é dada por:

$$Wf[n, s] = \sum_{m=0}^{N-1} f[m] \frac{1}{s} \psi^*\left[\frac{m-n}{s}\right]. \quad (2.3)$$

Para sua formulação em tempo discreto, Mallat (1989) define a DWT a partir da convolução paralela do sinal com dois filtros e posterior sub-amostragem por fator 2. Um filtro passa-baixas \mathbf{H} gera o vetor \mathbf{A} chamado de aproximação, enquanto um passa-altas \mathbf{G} gera o vetor \mathbf{D} chamado de detalhamento. Por efeito da sub-amostragem, os vetores \mathbf{A} e \mathbf{D} apresentam metade do número de amostras do sinal, sendo regularmente concatenados em um único vetor w com tamanho igual ao sinal original. Um diagrama de blocos da aplicação da DWT a um sinal qualquer $x[n]$ está exposto na Figura 3.

Figura 3 – Diagrama de blocos da transformada *wavelet* discreta.



Fonte: Adaptado de Mallat (1989).

A definição dos filtros depende de ψ e de uma função de escala ϕ , também chamada de *wavelet* pai. A relação entre as funções é dada pela equação:

$$|\hat{\phi}(\omega)|^2 = \int_1^{+\infty} \frac{|\hat{\psi}(s\omega)|^2}{s} ds, \quad (2.4)$$

onde $\hat{\phi}$ e $\hat{\psi}$ correspondem à transformada de Fourier de ϕ e ψ , respectivamente. A fase de ϕ pode ser definida arbitrariamente.

O filtro **H** depende da função ϕ , e é definido como $h[n] = \sqrt{2}\tilde{\phi}[-n]$, sendo $\tilde{\phi}$ a função ϕ normalizada. O filtro **G**, por sua vez, é definido de forma que $g[n] = (-1)^{1-n}h[1-n]$ (MALLAT, 1989).

Como exemplo desta metodologia de transformação, a função base Haar (Equação 2.5) será utilizada para decompor o sinal *blocks* (Figura 2).

$$\psi(x) = \begin{cases} 1 & , \text{ se } 0 \leq x < 0,5 \\ -1 & , \text{ se } 0,5 \leq x \leq 1 \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.5)$$

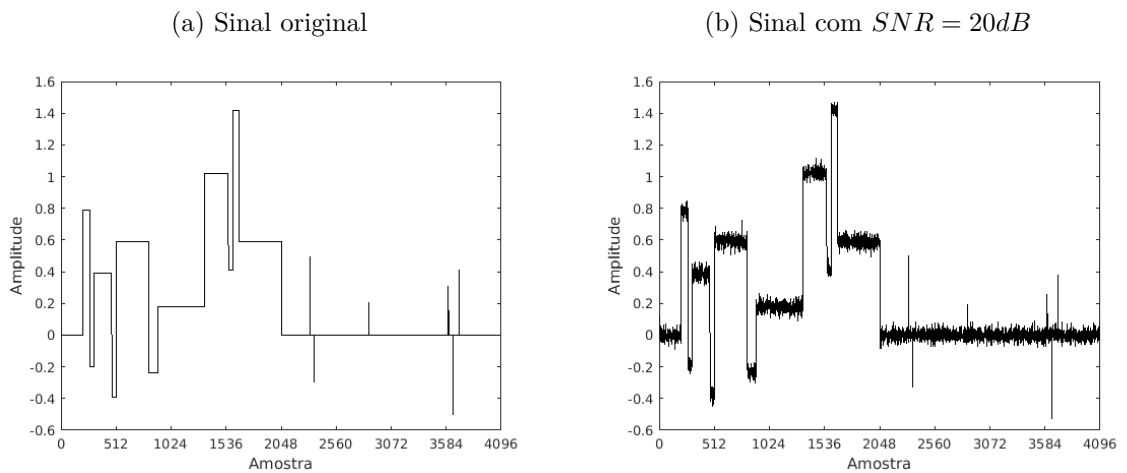
Com a utilização da Equação 2.4, a função de escala para a transformada Haar é definida como:

$$\phi(x) = \begin{cases} 1 & , \text{ se } 0 \leq x \leq 1 \\ 0 & , \text{ caso contrário} \end{cases}, \quad (2.6)$$

de forma que os coeficientes dos filtros **H** e **G** são definidos como $h = [\sqrt{2}/2 \quad \sqrt{2}/2]$ e $g = [-\sqrt{2}/2 \quad \sqrt{2}/2]$.

Assim, os coeficientes *wavelet* (vetores de aproximação e detalhamento concatenados nesta ordem) do sinal estão expostos na Figura 4.

Figura 4 – Sinal *blocks* original e com ruído decomposto na função Haar.



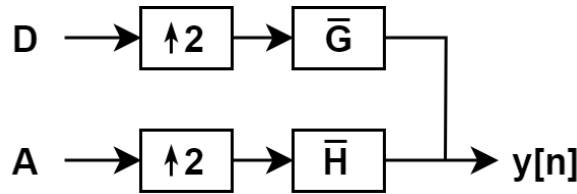
Fonte: O próprio autor

Em ambas as figuras, o vetor de aproximação (**A**) e o vetor de detalhamento (**D**) são representados pelos coeficientes nas amostras dos intervalos $[1 \ 2048]$ e $[2049 \ 4096]$, respectivamente. É possível observar que os 6 componentes não nulos do vetor **D** do sinal

original (Figura 4a) apresentam coeficientes no domínio *wavelet* superiores aos coeficientes do ruído, o que permite um aumento da SNR do sinal após as etapas de limiarização e transformação inversa.

A aplicação da transformada inversa consiste na convolução dos vetores de aproximação e detalhamento com dois filtros precedida de super-amostragem por fator 2 (Figura 5). Os filtros de reconstrução $\bar{\mathbf{H}}$ e $\bar{\mathbf{G}}$ são definidos como $\bar{h}[n] = h[-n]$ e $\bar{g}[n] = g[-n]$. A Figura 5 apresenta um diagrama de blocos da aplicação da IDWT em dois vetores \mathbf{A} e \mathbf{D} quaisquer.

Figura 5 – Diagrama de blocos da transformada *wavelet* discreta inversa.



Fonte: Adaptado de Mallat (1989).

2.1.1 Forma Matricial

A DWT, formulada conforme Figura 3, pode ser representada como uma multiplicação matricial linear, de forma que:

$$w = \mathcal{W}x. \quad (2.7)$$

Para um sinal x de tamanho N , a matriz de transformação \mathcal{W} será $N \times N$, e seus elementos dependem diretamente dos filtros \mathbf{H} e \mathbf{G} . A título de exemplo, a matriz de transformação com base nos filtros da função Haar é dada por:

$$\mathcal{W} = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \sqrt{2}/2 & \sqrt{2}/2 & \dots & 0 & 0 \\ \vdots & & & \ddots & & & \\ 0 & 0 & 0 & 0 & \dots & \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \sqrt{2}/2 & -\sqrt{2}/2 & \dots & 0 & 0 \\ \vdots & & & \ddots & & & \\ 0 & 0 & 0 & 0 & \dots & \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix}. \quad (2.8)$$

A formulação de Mallat (1989), aqui apresentada, resulta em matrizes ortogonais, portanto:

$$x = \mathcal{W}^T w. \quad (2.9)$$

Definida a norma l_2 de um vetor x qualquer como:

$$\|x\|_2 = \sqrt{\sum_{k=1}^N x[k]^2} = \sqrt{x^T x}, \quad (2.10)$$

a norma l_2 de um vetor de coeficientes *wavelet* é dada por:

$$\|w\|_2 = \sqrt{w^T w} = \sqrt{x^T \mathcal{W}^T \mathcal{W} x}. \quad (2.11)$$

Como a transformação é ortogonal, o produto $\mathcal{W}^T \mathcal{W}$ é igual a matriz identidade. Portanto:

$$\|w\|_2 = \sqrt{x^T x} = \|x\|_2. \quad (2.12)$$

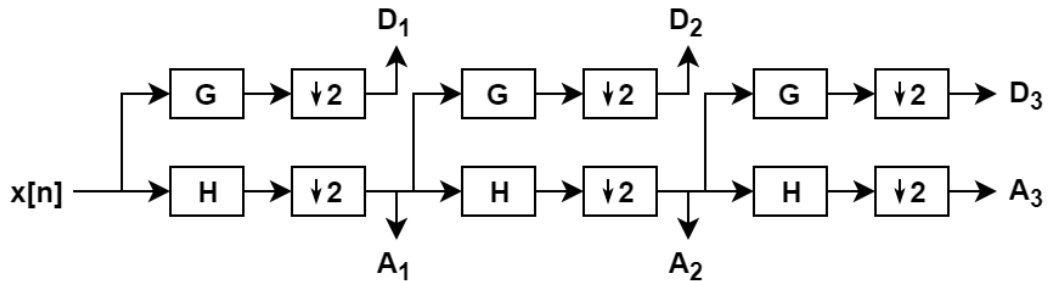
Assim, a norma l_2 da representação de um sinal qualquer por uma transformação ortogonal é constante e igual a energia do sinal no domínio temporal.

Devido à propriedade de ortogonalidade, uma realização do ruído branco é transformada em ruído branco (DONOHO; JOHNSTONE, 1995). Para o caso de ruído branco com média zero (gaussiano), sua potência é proporcional ao quadrado da norma l_2 . Portanto, a imagem da transformação de um ruído branco gaussiano apresenta a mesma potência do ruído no domínio do tempo.

2.1.2 Análise Multiresolução

Segundo Mallat (1989), a decomposição de um sinal em níveis diferentes de resolução consiste em aplicações sucessivas da DWT nos vetores de aproximação de cada etapa anterior. O número de transformações utilizadas é denominado escala. Tendo como exemplo a decomposição na 3ª escala, a Figura 6 expressa o diagrama de blocos da transformação.

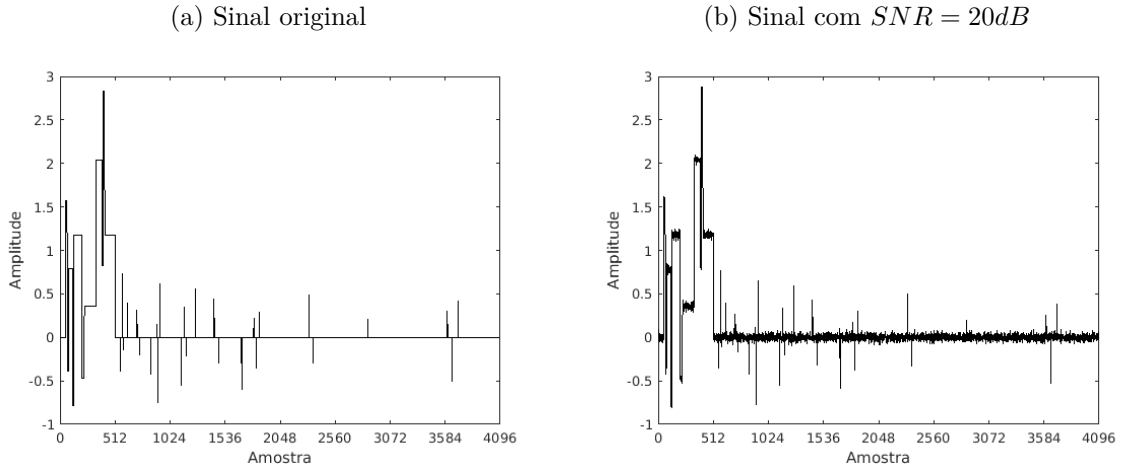
Figura 6 – Diagrama de blocos da transformada *wavelet* discreta na escala 3.



Fonte: O próprio autor

Se o sinal $x[n]$ possui tamanho N , os sinais D_1 , D_2 , D_3 e A_3 tem $N/2$, $N/4$, $N/8$ e $N/8$ amostras, respectivamente. Portanto, se representados em um único vetor concatenado w , apresentam tamanho igual ao sinal original. No Apêndice A o Algoritmo 1 resume a transformada *wavelet* discreta multiresolução.

A Figura 7 apresenta o vetor w resultante da DWT do sinal *blocks* em escala 3. Figura 7 – Sinal *blocks* original e com ruído decomposto na função Haar em escala 3.



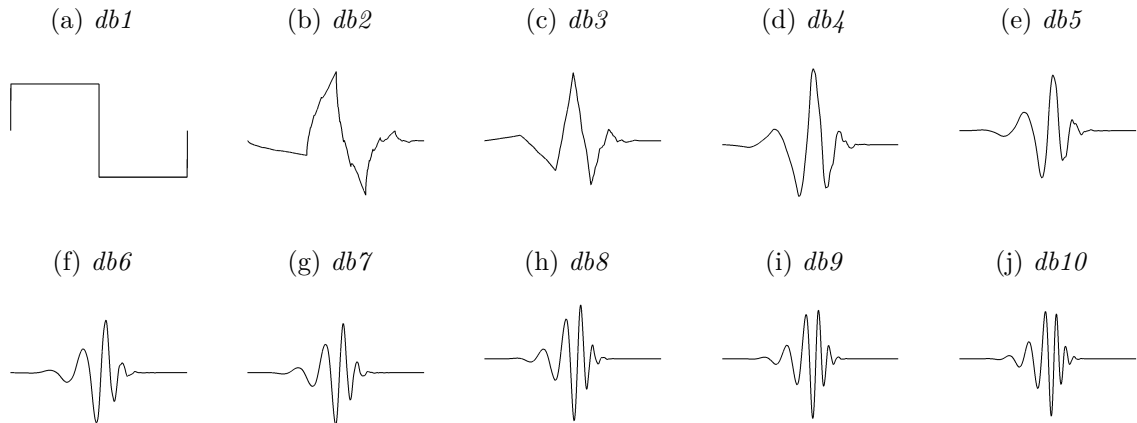
Fonte: O próprio autor

Em ambas as figuras, os vetores \mathbf{A}_3 , \mathbf{D}_3 , \mathbf{D}_2 e \mathbf{D}_1 são representados pelos coeficientes nas amostras dos intervalos $[1\ 512]$, $[513\ 1024]$, $[1025\ 2048]$ e $[2049\ 4096]$, respectivamente. Semelhantemente à decomposição na primeira escala, todos os coeficientes de detalhamento (vetores \mathbf{D}_1 , \mathbf{D}_2 e \mathbf{D}_3) do sinal original possuem amplitude superior aos coeficientes do ruído.

2.1.3 Bases Ortogonais

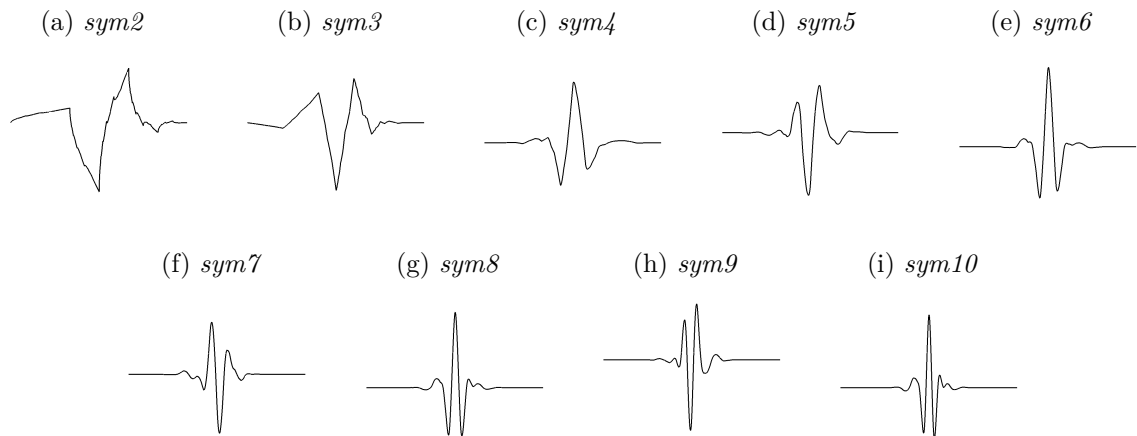
Assim como a função Haar (Equação 2.5), diversas funções *wavelet* podem ser usadas no processamento de sinais. Como a transformação tem o objetivo de representar um sinal com poucos coeficientes não-nulos, o projeto de uma base *wavelet* consiste na definição de uma função com suporte mínimo. Em conjunto a este requisito, outras restrições podem ser impostas, de forma que objetivos distintos geram diferentes famílias de bases. O conjunto de bases ortogonais é formado pelas famílias Daubechies, Symlet e Coiflet (todas propostas por Daubechies) e pela função discreta de Meyer (MALLAT, 2008).

A família Daubechies é formada por funções que, dado um número de momentos nulos p , possuem o menor suporte possível. Cada base é, então, referida pela abreviatura 'db' acrescida do número de momentos nulos, como db3 para $p = 3$. Para $p = 1$, a função com o menor suporte é a função Haar e, portanto, pode ser também chamada de db1. A Figura 8 apresenta as bases da família Daubechies com até 10 momentos nulos.

Figura 8 – Funções Daubechies para $1 \leq p \leq 10$.

Fonte: O próprio autor

Ao propor as funções de mínimo suporte, Daubechies (1992) demonstrou que a única função de suporte compacto que apresenta filtro com fase linear é a Haar (db1). Para $p > 1$, a assimetria das funções *wavelet* gera filtros com fase não-linear. Foram propostas, então, variações das funções da família Daubechies com o objetivo de produzir filtros com fase quase linear. A nova família de bases, devido a sua quase simetria, é denominada Symlet. Cada base é referida pela abreviatura 'sym' acrescida do número de momentos nulos e, como db1 apresenta fase linear, a primeira função da família é a sym2. A Figura 9 apresenta as bases da família Symlet com até 10 momentos nulos.

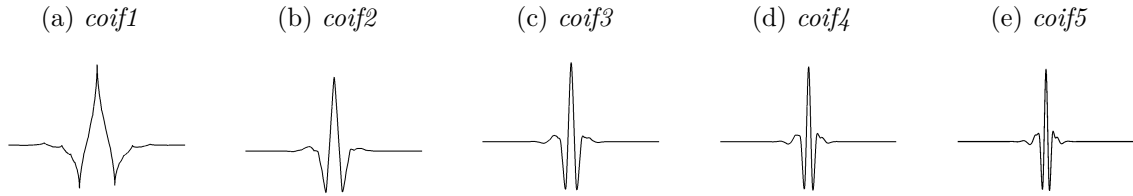
Figura 9 – Funções Symlets para $2 \leq p \leq 10$.

Fonte: O próprio autor

Devido a uma sugestão do matemático Ronald Coifman, Daubechies (1992) propôs também bases cujas funções de escala, assim como as *wavelets*, possuem p momentos nulos. A família foi denominada Coiflet, e cada base é referida pela abreviatura 'coif' acrescida

no número de momentos nulos. A Figura 10 apresenta as bases da família Coiflet com até 5 momentos nulos.

Figura 10 – Funções Coiflet para $1 \leq p \leq 5$.



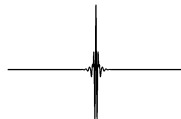
Fonte: O próprio autor

A função discreta de Meyer, referida como 'dmey', é definida como a base cuja transformada de Fourier do filtro \mathbf{H} é dada por:

$$\hat{h}(\omega) = \begin{cases} 0 & , \text{ se } -\pi \leq \omega < -2\pi/3 \\ \sqrt{2} & , \text{ se } -\pi/3 \leq \omega \leq \pi/3 \\ 0 & , \text{ se } 2\pi/3 < \omega \leq \pi \end{cases} \quad (2.13)$$

A transição entre as bandas deve satisfazer $|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2$ e garantir que a função seja infinitamente derivável. A função resultante, portanto, é conforme Figura 11.

Figura 11 – Função discreta de Meyer.



Fonte: O próprio autor

As funções descritas nesta seção compõe o conjunto de bases ortogonais disponíveis para a transformada *wavelet* de um sinal qualquer. No *software* Matlab®, utilizado neste trabalho, as bases db1 até db45, sym2 até sym45, coif1 até coif5 e dmey estão disponíveis, totalizando 95 bases. Como será observado na Subseção 3.1.2, o custo computacional da decomposição de um sinal aumenta com o número de momentos nulos (p) da base, em especial na família Symlet. Assim, as funções das famílias Daubechies e Symlet são comumente usadas com $p \leq 10$.

2.2 Limiarização

A etapa de transformação é seguida pela definição de um limiar que será aplicado ao sinal, com o intento de eliminar os coeficientes do ruído e preservar os coeficientes

do sinal original, simultaneamente. Nesta etapa, a escolha de uma técnica para definir o limiar é a prioridade, visto que a posterior aplicação do limiar é um processo simples, como será visto ao final desta seção. Para exemplificar os métodos discutidos, o sinal *blocks* decomposto pela função Haar na escala 1 (Figura 4) será utilizado.

Para o caso do sinal em análise, a menor amplitude de um coeficiente do sinal original é 0,206, enquanto a maior amplitude de um coeficiente do ruído é 0,099. Assim, qualquer limiar entre estes valores resultará na maior restauração para esta situação. Porém, nem sempre a separação de amplitude entre o sinal e o ruído acontece como neste caso, podendo ocorrer sobreposição de amplitude. Nestes casos, a eliminação completa dos coeficientes do ruído resultaria na eliminação de componentes significativos do sinal e, portanto, é preferível que o limiar seja inferior ao coeficiente de menor amplitude do sinal para que, apesar da preservação de coeficientes ruidosos, as características do sinal não sejam distorcidas.

A técnica de limiarização mais simples é o limiar universal, definido como (DONOHO; JOHNSTONE, 1994):

$$t = \sqrt{2\ln(N)}, \quad (2.14)$$

onde N é o número de amostras do sinal.

Para o sinal *blocks* (Figura 4) o limiar universal é $t = \sqrt{2\ln(4096/2)} \approx 4,08$, o que é inadequado visto que é superior as amplitudes do sinal original. Assim, o limiar universal, apesar de simples, não é robusto, visto que depende exclusivamente do número de amostras do sinal.

2.2.1 Estimador Não-enviesado de Risco de Stein

A técnica mais amplamente utilizada para a definição de limiares é o estimador de Stein (SURE), que estima de forma não-enviesada o erro de um estimador fixo qualquer (STEIN, 1981). Assim, para qualquer limiar t , o erro quadrático médio pode ser estimado para que o limiar com o menor erro seja selecionado. É gerado, portanto, um problema de otimização, no qual o limiar universal é utilizado para limitar o espaço de busca.

Para um limiar t e um vetor de coeficientes w fixos, o estimador de Stein é dado por:

$$SURE(t, w) = N - 2 \cdot \#\{i : |w[i]| \leq t\} + \sum_{i=1}^N (\min(|w[i]|, t)^2), \quad (2.15)$$

onde $\#\{i : |w[i]| \leq t\}$ indica o número de elementos do conjunto de índices i tal que $\{|w[i]| \leq t\}$.

A partir da Equação 2.15 e utilizando o limiar universal como o limite superior do espaço de busca, o limiar t_s que minimiza o risco de Stein é dado por (DONOHO;

JOHNSTONE, 1995):

$$t_s = \operatorname{argmin}_{0 \leq t \leq \sqrt{2 \ln(N)}} \operatorname{SURE}(t, w). \quad (2.16)$$

A resolução do problema de otimização exposto na Equação 2.16 está descrita no Algoritmo 2 do Apêndice A.

Ao aplicar o método do estimador de Stein ao sinal *blocks*, o limiar que minimiza o risco estimado é 0,099 e isto revela duas qualidades do método. Em primeiro lugar, o limiar definido está na faixa esperada para a situação em análise e, portanto, irá resultar na melhor reconstrução possível para a transformada Haar em primeira escala. Em segundo lugar, o valor se encontra no limite inferior da faixa, e portanto não há risco de distorção dos coeficientes do sinal.

2.2.2 Aplicação de Limiares

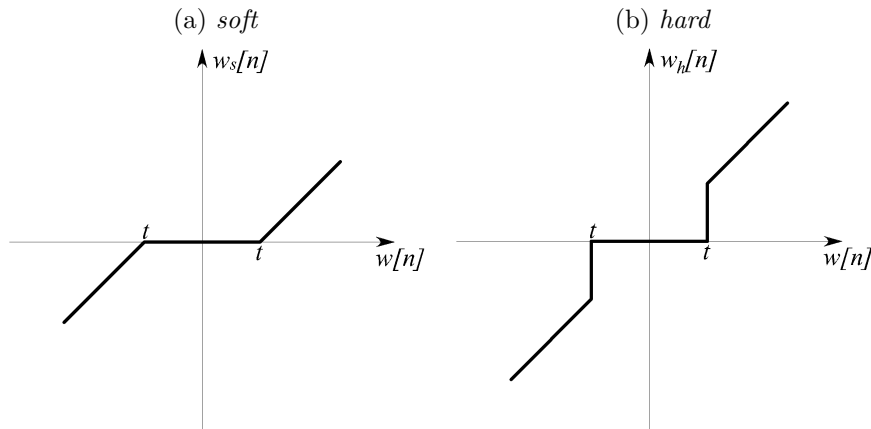
Após a definição de um limiar t , este deve ser aplicado ao sinal de forma que coeficientes com amplitudes inferiores a t sejam eliminados. Há dois modos de aplicação de limiares: suave (*soft thresholding*) e dura (*hard thresholding*). A partir de coeficientes *wavelet* $w[n]$, os novos coeficientes $w_s[n]$ e $w_h[n]$ por limiarização suave e dura, respectivamente, são definidos por (DONOHO; JOHNSTONE, 1994):

$$w_s[n] = \begin{cases} \operatorname{sgn}(w[n])(|w[n]| - t) & , \text{ se } |w[n]| \geq t \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.17)$$

$$w_h[n] = \begin{cases} w[n] & , \text{ se } |w[n]| \geq t \\ 0 & , \text{ caso contrário} \end{cases}. \quad (2.18)$$

Na Equação 2.17, $\operatorname{sgn}(w[n])$ representa o sinal do coeficiente $w[n]$. As funções de ativação para as limiarizações suave e dura estão expostas na Figura 12.

Figura 12 – Funções de ativação de limiarização.

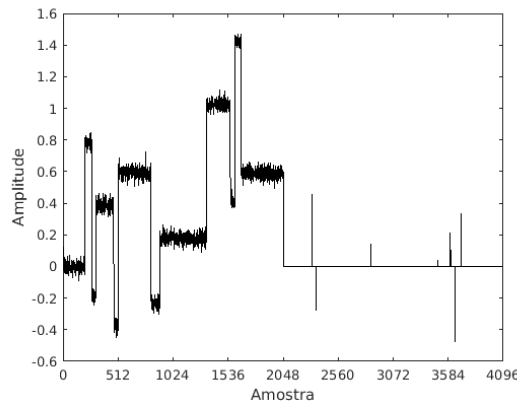


Fonte: O próprio autor

Segundo Donoho e Johnstone (1995), a limiarização suave é preferível devido a vantagens estatísticas como continuidade, simplicidade de cálculo e prova do estimador de Stein. No Apêndice A o Algoritmo 3 resume o *wavelet shrinkage* com o uso da limiarização suave.

Ao aplicar a limiarização suave no sinal *blocks* com $t = 0,099$, a representação do sinal é conforme Figura 13.

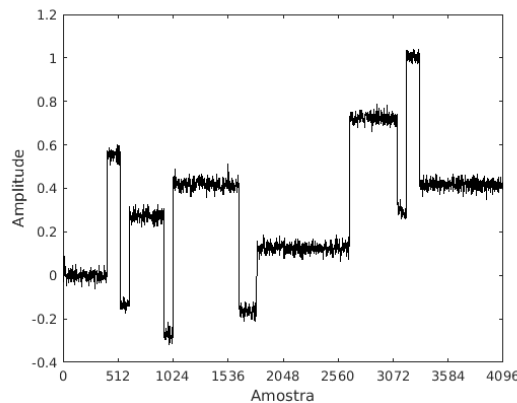
Figura 13 – Sinal *blocks* decomposto na função Haar após limiarização.



Fonte: O próprio autor

É possível observar que o vetor de aproximação (amostras 1 a 2048) é semelhante à decomposição do sinal ruidoso (Figura 4b), enquanto o vetor de detalhamento (amostras 2049 a 4096) é semelhante à decomposição do sinal original (Figura 4a). Assim, após a aplicação da transformada inversa (Figura 5), o sinal sintetizado apresenta $SNR = 22,6dB$ (Figura 14).

Figura 14 – Sinal *blocks* recuperado por função Haar e SURE.



Fonte: O próprio autor

Apesar do pequeno ganho na relação sinal ruído, todas as bordas do sinal foram mantidas, o que não seria possível com a aplicação de um filtro passa-baixas clássico. Além

disso, o ruído foi atenuado mesmo em regiões de alta amplitude do sinal, o que não seria possível com a limiarização no domínio do tempo (*noise gate*).

2.3 Influência dos Parâmetros

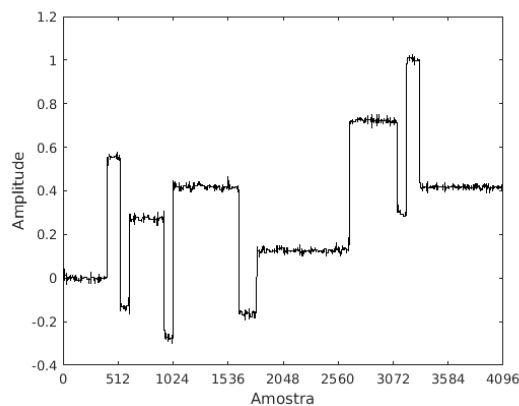
O método apresentado possui apenas dois parâmetros a serem definidos pelo usuário: base de transformação e escala de aplicação. Para analisar a influência destes parâmetros no desempenho do método, o sinal *blocks* com $SNR = 20\text{dB}$ foi decomposto por 6 bases diferentes em 5 níveis, totalizando 30 representações. Após a definição de um limiar (Algoritmo 2), sua aplicação de forma suave (Equação 2.17) e síntese do sinal, a SNR foi calculada para cada par base/nível. O resultado deste teste encontra-se na Tabela 1. A maior e a menor SNR's obtidas estão em negrito e itálico, respectivamente.

Tabela 1 – SNR (dB) do sinal *blocks* recuperado por 6 bases em 5 níveis distintos. O melhor e o pior resultados estão destacados em negrito e itálico, respectivamente.

Escala	haar	db2	db3	db4	coif1	coif2
1	22,6	22,7	22,7	<i>22,5</i>	22,6	22,6
2	25,4	25,2	25,1	25,0	25,2	25,1
3	27,5	27,0	26,9	26,5	27,1	26,7
4	28,9	27,9	27,6	27,2	28,2	27,5
5	29,6	28,4	27,9	27,3	28,7	27,7

A partir desse teste, é possível observar que o conjunto de parâmetros de transformação tem grande peso no desempenho do método, visto que o ganho, em dB, na relação sinal-ruído variou de 2,5 a 9,6. O melhor resultado foi obtido ao aplicar a função Haar na 5ª escala, e o sinal resultante está exposto na Figura 15.

Figura 15 – Sinal *blocks* recuperado por função Haar em escala 5.



Fonte: O próprio autor

Como a limiarização é aplicada ao vetor de detalhamento do sinal (Figura 13), o desempenho do método tende a aumentar com o aumento da escala (Tabela 1), visto que

mais amostras irão pertencer a este vetor. Porém, em função da complexidade e amostragem do sinal, o desempenho pode cair significativamente a partir de uma determinada escala. Isto é devido a sub-amostragem do vetor de aproximação a cada escala de aplicação, o que pode tornar a reconstrução menos eficiente.

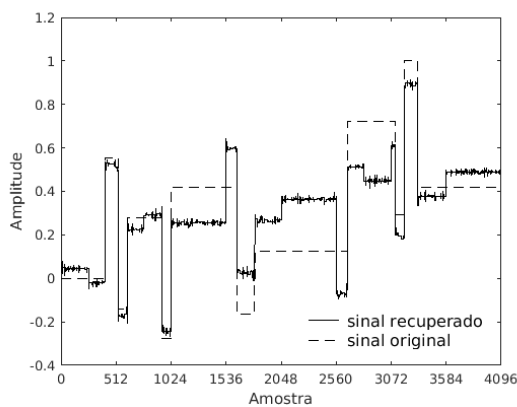
A título de exemplo, a SNR do sinal *blocks* após *denoising* pela função Haar nas dez primeiras escalas está apresentada na Tabela 2.

Tabela 2 – SNR (dB) do sinal *blocks* recuperado pela função Haar em 10 níveis distintos.

Escala	SNR
1	22,6
2	25,4
3	27,5
4	28,9
5	29,6
6	30,2
7	30,3
8	30,2
9	20,1
10	3,3

É possível observar que o desempenho máximo é obtido na escala 7, com redução significativa após a 8ª escala. O sinal recuperado no décimo nível (Figura 16) apresenta SNR muito inferior à condição inicial.

Figura 16 – Sinal *blocks* recuperado por função Haar em escala 10.



Fonte: O próprio autor

3 Materiais e Métodos

Dada a grande importância que a escolha da função *wavelet* e da escala de aplicação tem para o desempenho do *denoising* por DWT (Tabela 1), este capítulo irá apresentar a metodologia proposta para a definição automática destes parâmetros.

3.1 Esparsidade no Domínio *Wavelet*

Segundo Daubechies (1992), um sinal é melhor representado por uma função base semelhante a ele, sendo que uma boa representação é aquela na qual poucos coeficientes são necessários para caracterizar o sinal. Como a energia dos coeficientes *wavelet* de um sinal não varia com a base ou escala de aplicação (Equação 2.12), uma boa representação do sinal produz coeficientes com amplitudes mais altas que uma representação ruim. Assim, a diferença de amplitude entre os componentes do sinal original e do ruído tende a ser maior, característica importante para um *denoiser*.

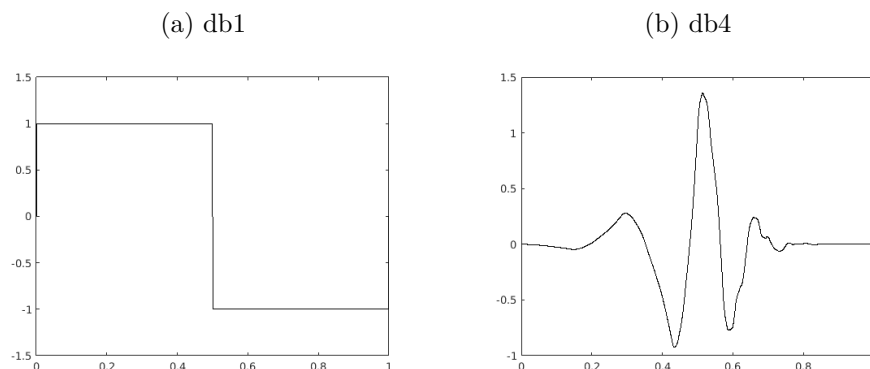
Matematicamente, como a transformada *wavelet* é linear, ao aplicar a Equação 2.7 na observação y de um sinal ruidoso (Equação 1.1), tem-se que:

$$\mathcal{W}y = \mathcal{W}x + \mathcal{W}e. \quad (3.1)$$

Como a transformação ortogonal do ruído branco ($\mathcal{W}e$) apresenta a mesma distribuição e potência que o ruído no domínio temporal (e), uma boa representação do sinal observado ($\mathcal{W}y$) tenderá a produzir maior diferença entre as amplitudes dos coeficientes do sinal original ($\mathcal{W}x$) e do ruído.

A título de exemplo, as funções db1 e db4 (Figura 17) podem ser usadas para decompor o sinal *blocks* (Figura 2).

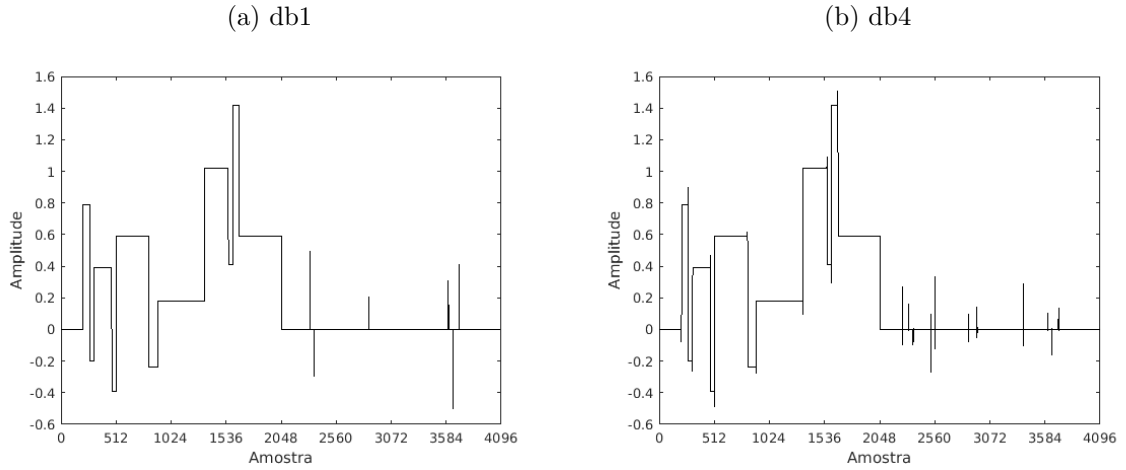
Figura 17 – Funções *wavelet* db1 e db4.



Fonte: O próprio autor

É possível observar que a função db1 tem maior semelhança com o sinal *blocks* que a função db4, e isto se evidencia nas representações do sinal (Figura 18).

Figura 18 – Sinal *blocks* decomposto nas funções db1 e db4.



Fonte: O próprio autor

A representação do sinal pela função db4 apresenta um maior número de elementos significativos que a representação pela função db1. Além disto, diversos coeficientes possuem amplitudes baixas que, a depender da SNR, podem estar na mesma faixa de amplitude dos coeficientes do ruído. Assim, a diferença de desempenho do *denoiser* utilizando essas duas funções (Tabela 1) pode ser atribuída à diferença na qualidade da representação do sinal.

3.1.1 Medidas de Esparsidade

Intuitivamente, uma representação esparsa de um sinal é aquela na qual um pequeno número de coeficientes contém uma grande proporção da energia total do sinal. Segundo Hurley e Rickard (2009), a métrica tradicional de esparsidade em diversos contextos é a norma l_0 , definida para um vetor de coeficientes w como:

$$\|w\|_0 = \#\{i : w[i] \neq 0\}. \quad (3.2)$$

A norma l_0 apresenta, porém, baixo desempenho na aferição da esparsidade de sinais ruidosos, visto que estes apresentam diversos coeficientes não-nulos de amplitude pouco significativa. Com o intuito de avaliar as diversas métricas de esparsidade existentes, Hurley e Rickard (2009) aplicaram seis critérios utilizados nas ciências econômicas na avaliação de distribuição de renda. Os critérios, expostos na Tabela 3, foram considerados como desejáveis para uma medida qualquer de esparsidade.

Tabela 3 – Critérios desejáveis de medidas de esparsidade.

Critério	Descrição
Robin Hood	Se um fator $\alpha > 0$ for subtraído de um elemento de alta amplitude e somado a um elemento de baixa amplitude, a medida de esparsidade deve diminuir
Escala	Se todos os elementos forem multiplicados por um fator $\alpha > 0$, a medida de esparsidade deve ser a mesma
Maré crescente	Se todos os elementos forem somados por um fator $\alpha > 0$, a medida de esparsidade deve diminuir
Clonagem	Se o vetor for concatenado com outro vetor idêntico ao original, a medida de esparsidade deve ser a mesma
Bill Gates	A medida que a amplitude de um elemento torna-se infinitamente grande, a medida de esparsidade deve tender a máxima possível
Bebês	Se um elemento com amplitude nula for concatenado ao vetor, a medida de esparsidade deve aumentar

Na análise de Hurley e Rickard (2009), duas medidas de esparsidade destacaram-se: a medida de Hoyer e o índice de Gini. Este foi a única medida livre de parâmetros a atender os seis critérios, enquanto a medida de Hoyer falha apenas em atender ao critério da clonagem. Segundo Hurley e Rickard (2009), este critério é discutível em certas aplicações. No caso do presente trabalho, o objetivo é a comparação da esparsidade de sinais com mesmo número de amostras e, portanto, o critério da clonagem não é significativo.

Sendo w um vetor de coeficientes de tamanho N , a medida de esparsidade de Hoyer $H(w)$ é dada por (HOYER, 2004):

$$H(w) = \frac{\sqrt{N} - \frac{\|w\|_1}{\|w\|_2}}{\sqrt{N} - 1}. \quad (3.3)$$

Para o cálculo do índice de Gini, o vetor deve ser ordenado. Sendo w_O o vetor w após ordenação, a esparsidade pelo índice de Gini é dada por (HURLEY; RICKARD, 2009):

$$G(x) = 1 - 2 \sum_{k=1}^N \left[\frac{w_O[k]}{\|w_O\|_1} \left(\frac{N - k + \frac{1}{2}}{N} \right) \right]. \quad (3.4)$$

Apesar da importância e eficiência do índice de Gini em quantificar a esparsidade de um sinal, a necessidade de ordenação implica em maior custo computacional, em especial em aplicações que requerem a síntese do sinal. Por este motivo, a medida de Hoyer foi utilizada neste trabalho como métrica objetiva de esparsidade.

A medida de Hoyer (Equação 3.3) depende diretamente da relação entre as normas l_1 e l_2 do vetor de coeficientes. A norma l_1 é definida como:

$$\|w\|_1 = \sum_{k=1}^N |w[k]|. \quad (3.5)$$

Se houver um único componente não-nulo em w , as normas l_1 e l_2 são iguais, o que resulta em $H(w) = 1$. Para qualquer outro caso, $\|w\|_1 > \|w\|_2$, o que torna a medida de

Hoyer inferior a 1. Portanto, a medida de Hoyer representa a esparsidade de um vetor com um número real de valor máximo 1 (esparsidade máxima). Para as representações da Figura 18, por exemplo, a medida de Hoyer é 0,969 para a decomposição em Haar e 0,936 para a decomposição em Daubechies 4.

Ao aplicar 2.12 na Equação 3.3, tem-se que:

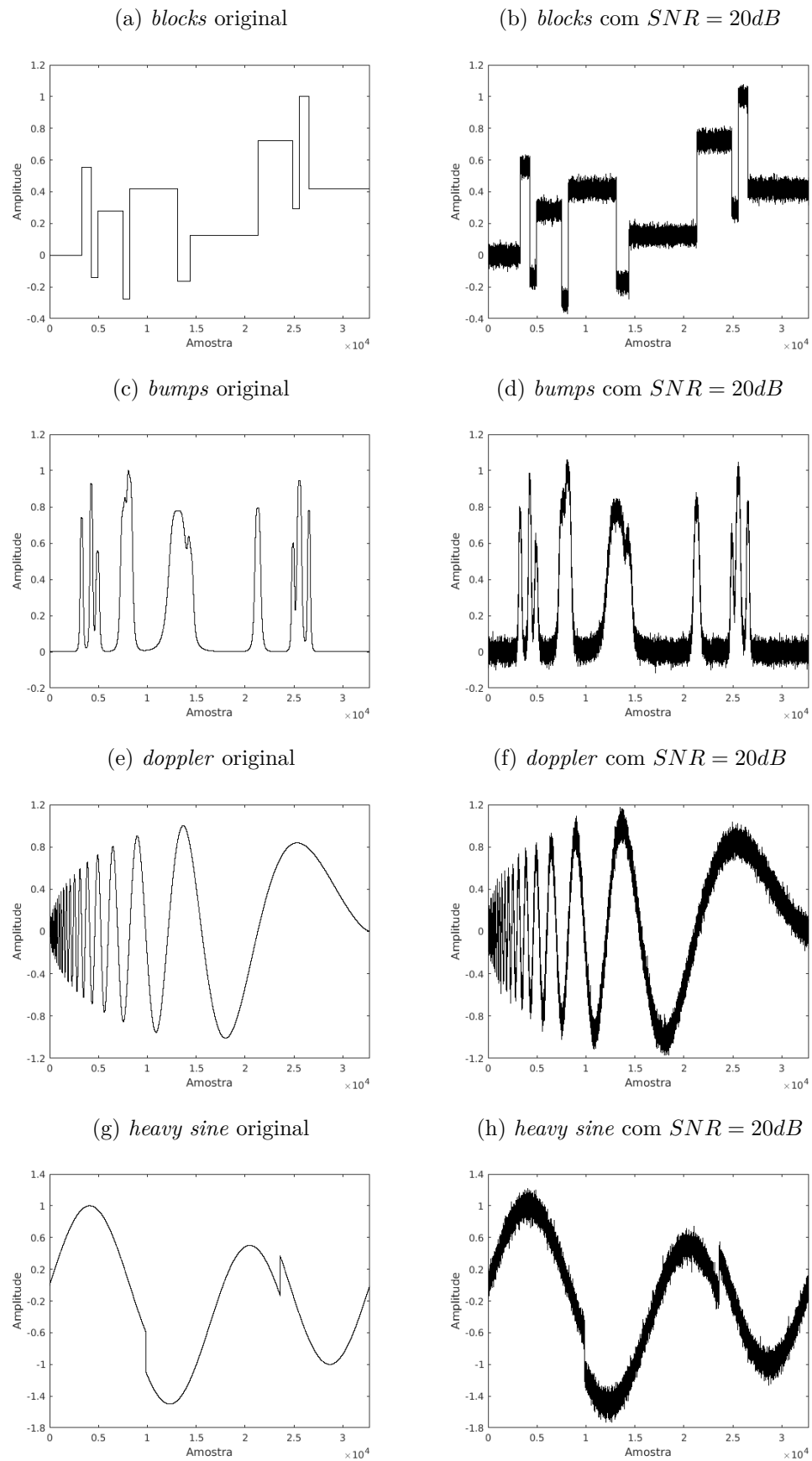
$$H(w) = \frac{\sqrt{N} - \frac{\|w\|_1}{\|x\|_2}}{\sqrt{N} - 1} = \frac{\sqrt{N}}{\sqrt{N} - 1} - \frac{\|w\|_1}{\|x\|_2(\sqrt{N} - 1)}. \quad (3.6)$$

Dado um sinal x , tanto seu tamanho (N) quanto sua norma l_2 ($\|x\|_2$) são constantes, portanto a medida de Hoyer da representação de x varia exclusivamente com a norma l_1 do vetor de coeficientes ($\|w\|_1$). Assim, dentre representações ortogonais de um sinal, aquela com maior esparsidade (no sentido da medida de Hoyer) é a representação com menor norma l_1 , o que reduz o custo computacional de comparar a esparsidade de duas ou mais transformações.

3.1.2 Esparsidade e Desempenho

Como observado na Tabela 1, o desempenho de um *denoiser* baseado na DWT é fortemente dependente da escolha de uma base e de uma escala de aplicação. Além disto, a Equação 3.1 indica a tendência de que, dado um sinal qualquer, um conjunto de parâmetros que produz uma representação mais esparsa que outro resulte em um maior ganho na SNR do sinal. Esta tendência, portanto, pode ser uma ferramenta para a escolha automática dos parâmetros de um *denoiser*.

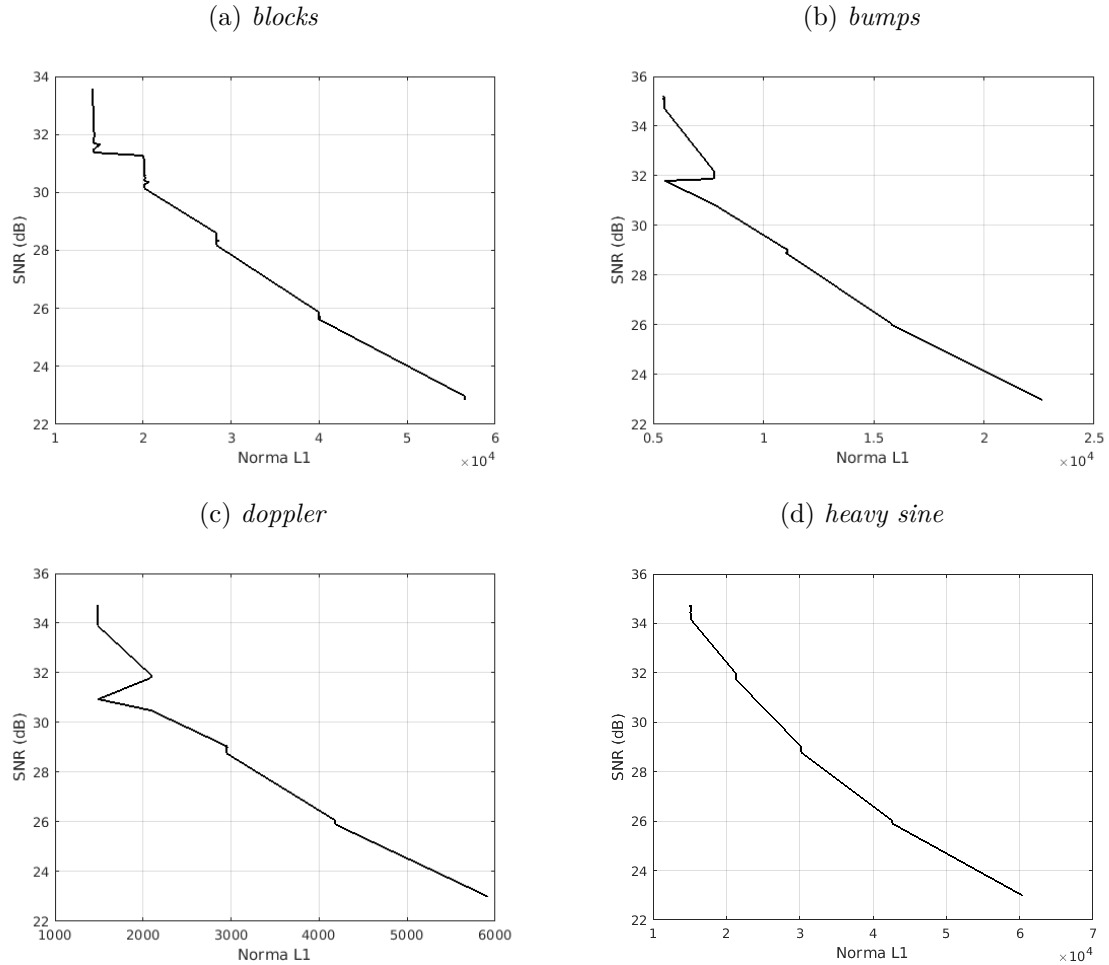
Para avaliar a relação entre a esparsidade da representação do sinal e o desempenho do *wavelet shrinkage*, quatro sinais foram utilizados: *blocks*, *bumps*, *doppler* e *heavy sine*, definidos por Donoho e Johnstone (1995), com 2^{15} amostras. Os sinais foram contaminados com ruído branco gaussiano de forma a ter $SNR = 20dB$ (Figura 19).

Figura 19 – Sinais *blocks*, *bumps*, *doppler* e *heavy sine* originais e com ruído.

Fonte: O próprio autor

Os sinais foram decompostos com as bases ortogonais disponíveis no Matlab® (Subseção 2.1.3) em 5 níveis, com $p \leq 10$ para as famílias Daubechies e Symlet. A norma l_1 da representação e SNR do sinal recuperado foram calculadas para cada par base/nível. Assim, os gráficos abaixo (Figura 20) representam o desempenho do *denoiser* (SNR) em função da esparsidade (norma l_1).

Figura 20 – Desempenho do *wavelet shrinkage* em função da esparsidade para os sinais *blocks*, *bumps*, *doppler* e *heavy sine*.



Fonte: O próprio autor

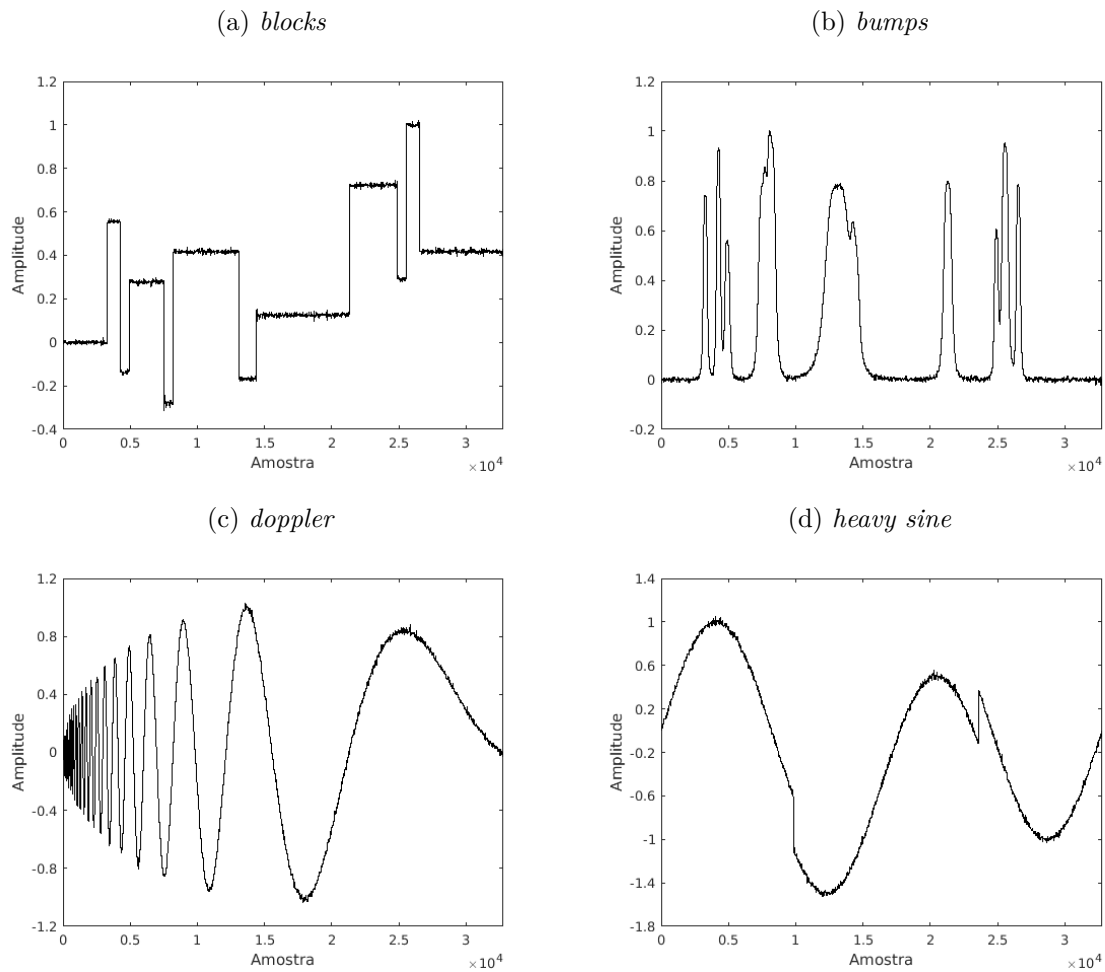
O número de momentos nulos das bases das famílias Daubechies e Symlet foi limitado em 10, pois foi observado que o tempo de processamento é grandemente elevado ao estender o teste para $p > 10$ sem que se observe uma significativa melhora de desempenho. Para o sinal *bumps*, um teste completo ($p \leq 45$) foi realizado com tempo de processamento de aproximadamente 68h20min, enquanto o teste com $p \leq 10$ foi realizado em aproximadamente 5s. No primeiro caso, o melhor desempenho obtido foi de 35,29dB, enquanto no segundo teste o melhor conjunto de parâmetros apresentou $SNR = 35,26$ dB. Assim, os testes foram realizados para $p \leq 10$, totalizando 23 bases, e apresentam 115 (23 bases em 5 escalas) pontos.

A partir do exposto na Figura 20, é possível observar novamente a grande influência do conjunto de parâmetros (base e nível) na recuperação do sinal, visto que a SNR dos sinais recuperados apresentou ampla faixa de valores. Além disto, é possível constatar que há forte relação entre a esparsidade da representação do sinal e o desempenho do algoritmo de *denoising*. Em todos os casos analisados, o conjunto de parâmetros que gerou o melhor resultado apresentou a menor norma l_1 . Porém, a presença de mínimos locais demonstra que dentre duas representações possíveis, não necessariamente a mais esparsa irá produzir a melhor recuperação do sinal, apesar da tendência global de isto ocorrer.

Ao comparar a Figura 20a com a Tabela 1, é possível observar também que o desempenho do *denoiser* aumenta com a elevação na taxa de amostragem. Isto justifica o uso de amostragens superiores à frequência de Nyquist (como 96kHz em sistemas de áudio) para o processamento de sinais. Porém, este não deve ser considerado um parâmetro de um algoritmo de *denoising*, visto que considera-se que o sinal já está amostrado.

A Figura 21 apresenta os sinais recuperados com maior SNR.

Figura 21 – Sinais *blocks*, *bumps*, *doppler* e *heavy sine* após *denoising*.



Fonte: O próprio autor

3.1.3 Minimização da Norma L1

Como observado na Subseção 3.1.2, existe uma forte tendência de que a melhor recuperação de um sinal por *wavelet shrinkage* seja aquela com representação mais esparsa. Propõe-se, portanto, escolher a base e escala de aplicação que minimizam a norma l_1 dos coeficientes *wavelet* do sinal. Esta seção irá discutir duas formas possíveis de determinar o conjunto de parâmetros.

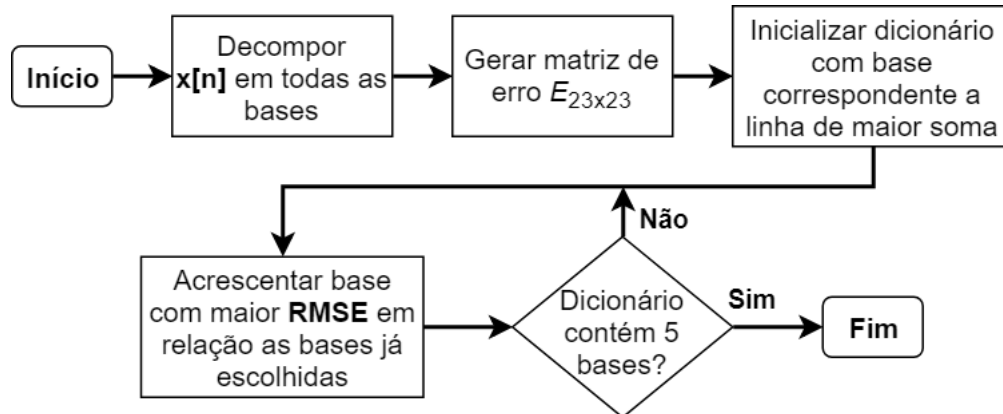
Em aplicações nas quais o custo computacional não é uma preocupação, os parâmetros do *denoiser* podem ser definidos pela busca exaustiva dentre as 23 bases ortogonais até uma escala máxima de busca definida pelo usuário. Para cada conjunto de parâmetros, a norma l_1 da representação do sinal é determinada, para que a decomposição com menor norma seja escolhida. Este método será referido como *l1shrinkage* completo, e está descrito no Algoritmo 4 do Apêndice A.

Algumas funções do dicionário, como db6 e db7, são semelhantes e geram, como consequência, representações parecidas. A presença de funções semelhantes no dicionário não oferece ganhos significativos no desempenho do *denoiser*, apesar do acréscimo no custo computacional. Esta observação motivou a proposta de um segundo método, que será referido como *l1shrinkage* rápido, no qual um dicionário conciso é utilizado. Para definir um novo dicionário, propõe-se escolher as 5 funções que geram as representações mais diferentes dentre as 23 do dicionário inicial. A dissemelhança entre duas representações $w_1[n]$ e $w_2[n]$ de tamanho N pode ser medida pela raiz do erro quadrático médio (RMSE), definido por:

$$RMSE(w_1, w_2) = \sqrt{\sum_{i=1}^N (w_1[i] - w_2[i])^2}. \quad (3.7)$$

Ao decompor um sinal nas bases e calcular o RMSE entre as $\binom{23}{2} = 253$ combinações de pares de representações, uma matriz de erro $E_{23 \times 23}$ é gerada, na qual o elemento e_{ij} corresponde ao erro entre as representações da base i e da base j . Assim, a soma dos elementos de uma linha i da matriz E indica a dissemelhança da representação $w_i[n]$ com relação ao conjunto de representações. A primeira base a compor o dicionário, portanto, é a base correspondente a linha cuja soma dos elementos é a maior. Cada nova base, então, é escolhida em função do erro de sua representação em relação as bases que já compõem o conjunto final. A Figura 22 resume a metodologia exposta.

Figura 22 – Metodologia para definição de dicionário conciso.



Fonte: O próprio autor.

Ao aplicar esta metodologia aos sinais da Figura 19, o dicionário conciso resultante é formado pelas bases db1, db10, sym4, coif5 e dmey. É possível observar a capacidade de representação deste dicionário pelo teste da Subseção 3.1.2, conforme exposto na Figura 20. Para o sinal *blocks*, o melhor desempenho foi obtido pela função db1, e para os demais sinais o dicionário conciso resultou em ganhos na SNR no máximo 0,1dB menores que o dicionário completo. O método rápido está descrito no Algoritmo 5 do Apêndice A.

3.2 Delineamento Experimental

Para comparar os métodos propostos na Subseção 3.1.3 com a metodologia proposta por Yu, Mallat e Bacry (2008) quanto ao desempenho, tempo de processamento e abrangência, foram utilizados seis sinais:

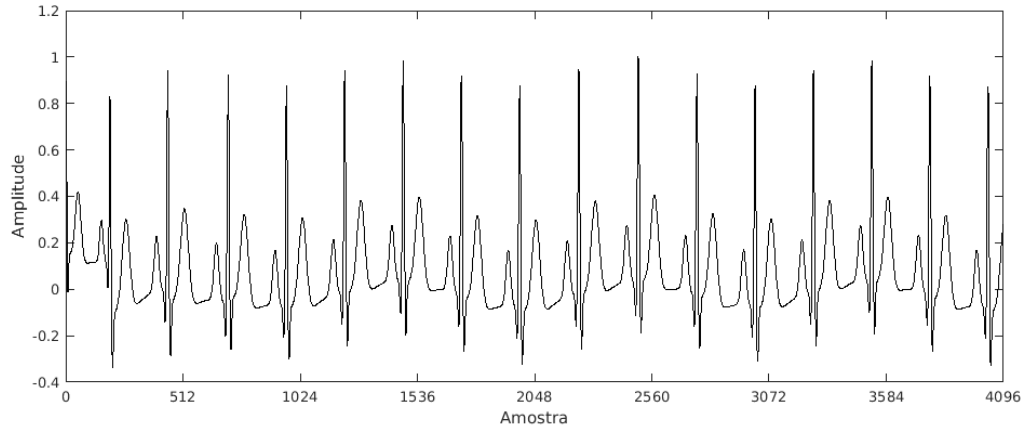
Tabela 4 – Sinais para teste.

Rótulo	Descrição
<i>blocks</i>	Função com saltos e amplitude constante entre saltos consecutivos
<i>bumps</i>	Somatório de lombadas com amplitudes e larguras diferentes
<i>doppler</i>	Sinal com frequência variante no tempo
<i>heavy sine</i>	Senoide com dois saltos
ECG	Eletrocardiograma gerado pelo ECGSYN da PhysioNet
Mozart	Trecho da <i>cadenza</i> de uma gravação do concerto de Mozart para oboé

Como um sinal de eletrocardiografia real sempre apresenta, mesmo que em pequena escala, ruído; optou-se pelo uso de um sinal simulado, que foi produzido pelo gerador de ondas ECGSYN da base de dados PhysioNet, descrito por McSharry et al. (2003). O ECGSYN sintetiza eletrocardiogramas com média e desvio padrão da frequência cardíaca, amostragem e morfologia (duração, amplitude e ritmo das ondas P, QRS e T) definidas pelo usuário, além da possibilidade da inclusão de patologias.

O sinal ECG foi gerado nas configurações padrão do ECGSYN, nas quais a amostragem é de 256 Hz e a frequência cardíaca é de 60 ± 1 bpm (Figura 23).

Figura 23 – Eletrocardiograma gerado pelo ECGSYN/PhysioNet.



Fonte: O próprio autor.

O sinal Mozart consiste no trecho da *cadenza* de uma gravação do concerto de Mozart para oboé e, portanto, apresenta apenas um instrumento sendo tocado. O sinal, cuja amostragem é de 11 kHz, foi utilizado por Yu, Mallat e Bacry (2008).

Cada sinal foi contaminado por ruído branco em diferentes potências, de forma a ter SNR's (em dB) iguais a 5, 10, 20 e 40. Para cada valor de SNR, vinte realizações de ruído branco foram utilizadas, totalizando 80 versões contaminadas para cada sinal. As técnicas de *l1shrinkage* completo, *l1shrinkage* rápido e *block thresholding* (descrito em (YU; MALLAT; BACRY, 2008)) foram aplicadas em cada sinal, com registro da SNR após o *denoising* e do tempo de processamento de cada técnica.

A escala máxima de busca foi definida como 6 para os sinais *blocks*, *bumps*, *doppler* e *heavy sine*. Para os sinais ECG e Mozart, a escala máxima de busca foi definida como 3 e 1, respectivamente, devido à maior complexidade destes sinais.

O método de *block thresholding* depende de três parâmetros: largura da janela de Hanning, estimativa da potência do ruído e frequência de amostragem, dos quais apenas os dois primeiras são parâmetros livres. Nos testes realizados, a janela de Hanning foi fixada em 100ms. A estimativa da potência do ruído foi realizada pela método de Garcia (2010).

Para avaliar diferença de desempenho entre os métodos *l1shrinkage* completo e rápido, foi aplicado um teste estatístico com a hipótese nula de que a diferença entre as SNRs obtidas pelos dois métodos apresenta distribuição normal com mediana zero, conforme descrito no Apêndice B. O intervalo de confiança assumido foi de 5%.

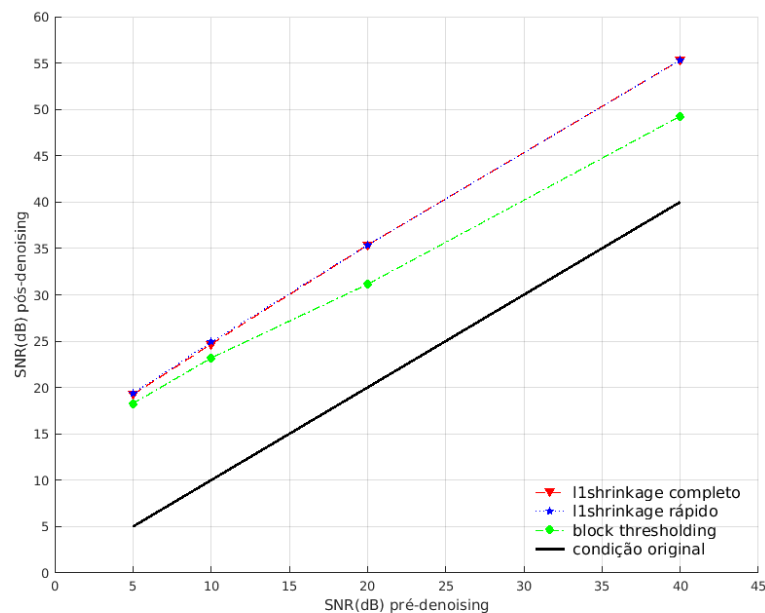
4 Resultados e Discussões

Este capítulo apresenta os resultados dos testes descritos na Seção 3.2, que consistem na aplicação das técnicas *l1shrinkage* completo, *l1shrinkage* rápido e *block thresholding* em diferentes sinais com diferentes SNRs. Para cada sinal em uma dada SNR, 20 experimentos com cada técnica foram realizados com a finalidade de registrar o desempenho e o tempo de processamento. Os testes resultaram em baixo desvio padrão, tanto em termos da SNR do sinal recuperado quanto de tempo de processamento. Assim, os gráficos aqui expostos apresentam unicamente a média dos experimentos para cada técnica em cada SNR inicial. O desvio padrão de cada teste, assim como a média, estão expostas em tabelas.

4.1 Relação Sinal-Ruído

A Figura 24 e a Tabela 5 apresentam a SNR do sinal recuperado em função da SNR inicial para o sinal *blocks*.

Figura 24 – Desempenho médio para o sinal *blocks*.



Fonte: O próprio autor

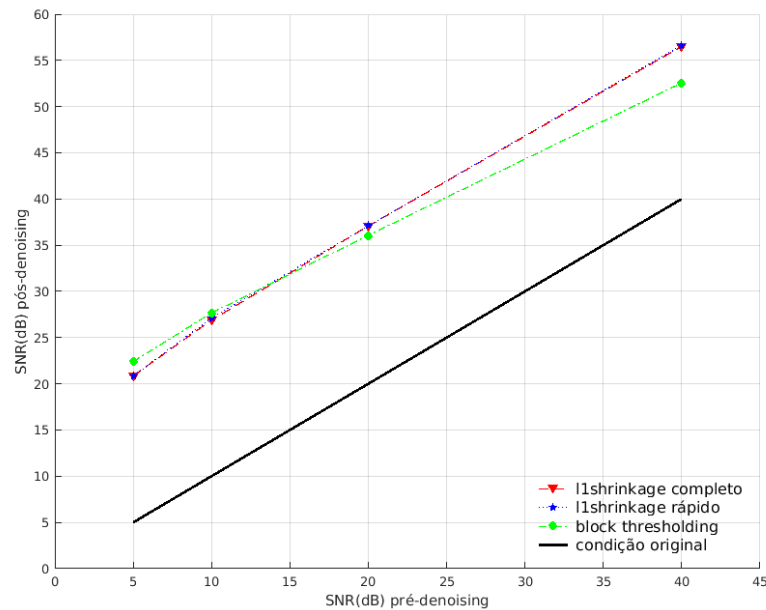
Na Figura 24, é possível observar que as técnicas baseadas na transformada *wavelet* apresentaram resultados superiores ao *block thresholding*, principalmente para SNR's mais altas. Enquanto o ganho, em dB, no *l1shrinkage* foi aproximadamente constante independente da condição inicial, houve uma queda no desempenho do *block thresholding*

Tabela 5 – SNR média (dB) do sinal *blocks*.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	19,2 \pm 0,3	19,3 \pm 0,3	18,3 \pm 0,2
10,0	24,7 \pm 0,3	24,9 \pm 0,2	23,2 \pm 0,2
20,0	35,3 \pm 0,2	35,3 \pm 0,2	31,1 \pm 0,1
40,0	55,3 \pm 0,3	55,3 \pm 0,3	49,2 \pm 0,1

com o aumento da SNR inicial. A diferença entre *l1shrinkage* completo e *l1shrinkage* rápido não foi significativa, especialmente nos casos com $SNR \geq 20dB$, nos quais as duas técnicas selecionaram a mesma base e escala em todos os testes. Todas as técnicas foram consistentes e apresentaram baixo desvio padrão.

A Figura 25 e a Tabela 6 apresentam a SNR do sinal recuperado em função da SNR inicial para o sinal *bumps*.

Figura 25 – Desempenho médio para o sinal *bumps*.

Fonte: O próprio autor

Tabela 6 – SNR média (dB) do sinal *bumps*.

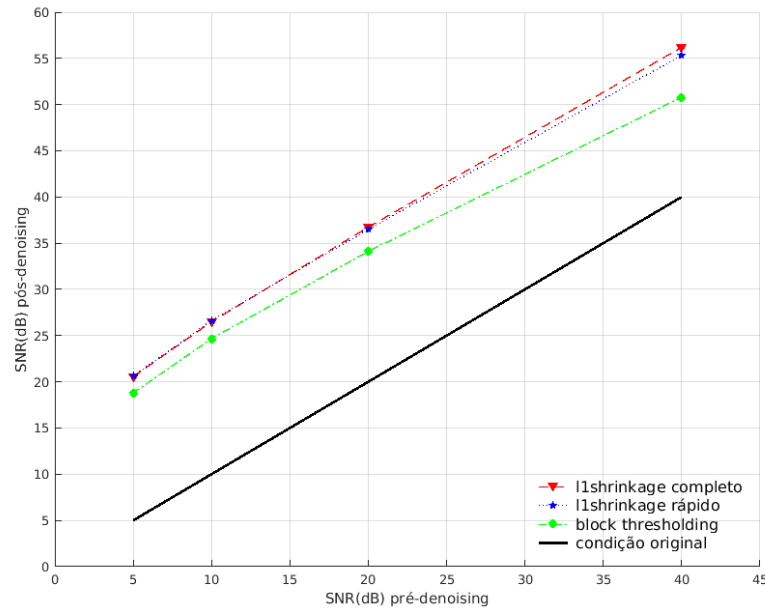
SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	20,9 \pm 0,3	20,8 \pm 0,7	22,4 \pm 0,2
10,0	26,9 \pm 0,4	27,1 \pm 0,4	27,7 \pm 0,2
20,0	37,1 \pm 0,4	37,1 \pm 0,4	36,1 \pm 0,2
40,0	56,5 \pm 0,3	56,6 \pm 0,3	52,6 \pm 0,1

Assim como no sinal *blocks*, é possível observar (Figura 25) a tendência do *block thresholding* de apresentar melhor desempenho em SNR iniciais mais baixas, enquanto este

é aproximadamente constante nas técnicas baseadas na DWT. A diferença de desempenho entre *l1shrinkage* completo e rápido não foi significativa.

A Figura 26 e a Tabela 7 apresentam a SNR do sinal recuperado em função da SNR inicial para o sinal *doppler*.

Figura 26 – Desempenho médio para o sinal *doppler*.



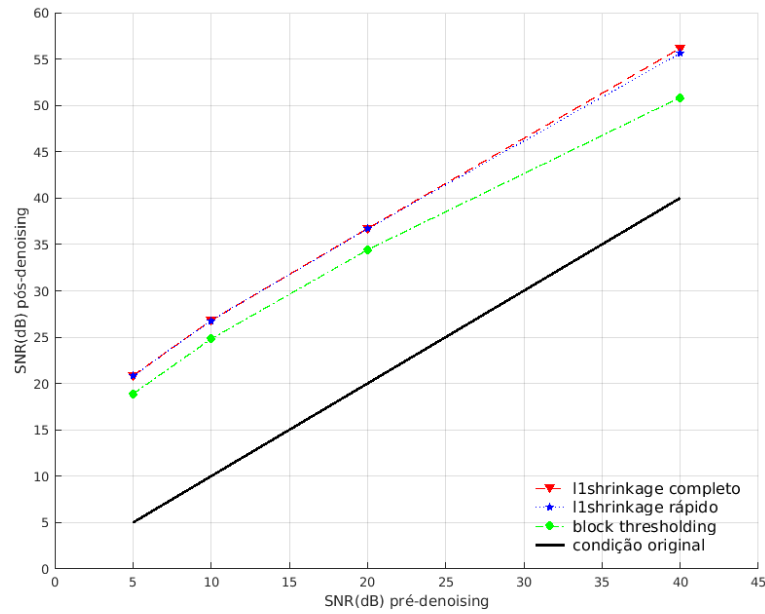
Fonte: O próprio autor

Tabela 7 – SNR média (dB) do sinal *doppler*.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	20,6 ± 0,4	20,6 ± 0,5	18,8 ± 0,2
10,0	26,5 ± 0,4	26,6 ± 0,3	24,7 ± 0,1
20,0	36,7 ± 0,4	36,5 ± 0,2	34,1 ± 0,2
40,0	56,2 ± 0,3	55,3 ± 0,5	50,8 ± 0,1

Na Figura 26, é possível observar que houve diferença significativa entre os desempenhos do *l1shrinkage* completo e rápido para $SNR = 40dB$, porém nos demais casos os dois métodos apresentaram resultados semelhantes. O *denoising* por *block thresholding* apresentou desempenho inferior às demais técnicas para todos os casos.

A Figura 27 e a Tabela 8 apresentam a SNR do sinal recuperado em função da SNR inicial para o sinal *heavy sine*.

Figura 27 – Desempenho médio para o sinal *heavy sine*.

Fonte: O próprio autor

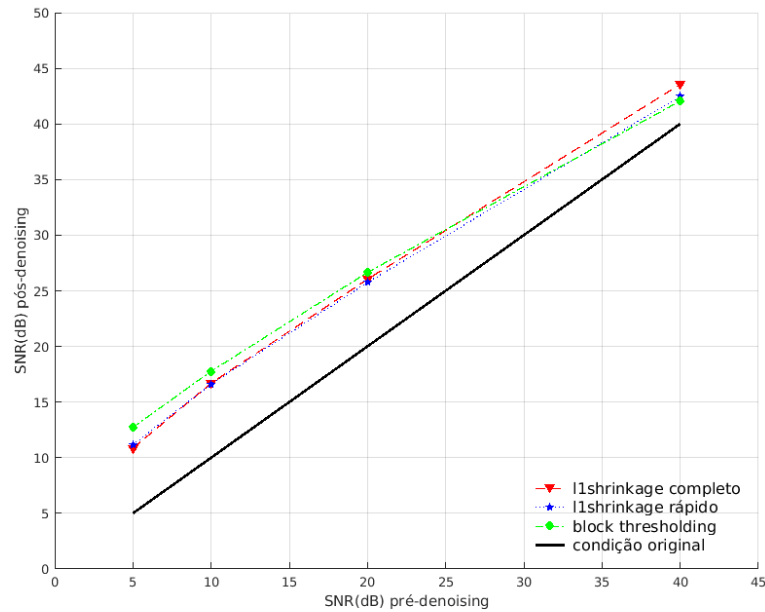
Tabela 8 – SNR média (dB) do sinal *heavy sine*.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	20,8 ± 0,3	20,8 ± 0,3	18,9 ± 0,2
10,0	26,8 ± 0,3	26,8 ± 0,3	24,8 ± 0,2
20,0	36,7 ± 0,3	36,7 ± 0,4	34,4 ± 0,2
40,0	56,2 ± 0,2	55,6 ± 0,2	50,9 ± 0,1

Assim como no sinal *doppler*, as duas técnicas baseadas na transformada *wavelet* no sinal *heavy sine* (Figura 27) apresentaram diferença significativa de desempenho apenas nos casos para $SNR = 40dB$.

A Figura 28 e a Tabela 9 apresentam a SNR do sinal recuperado em função da SNR inicial para o sinal ECG.

Figura 28 – Desempenho médio para o sinal ECG.



Fonte: O próprio autor

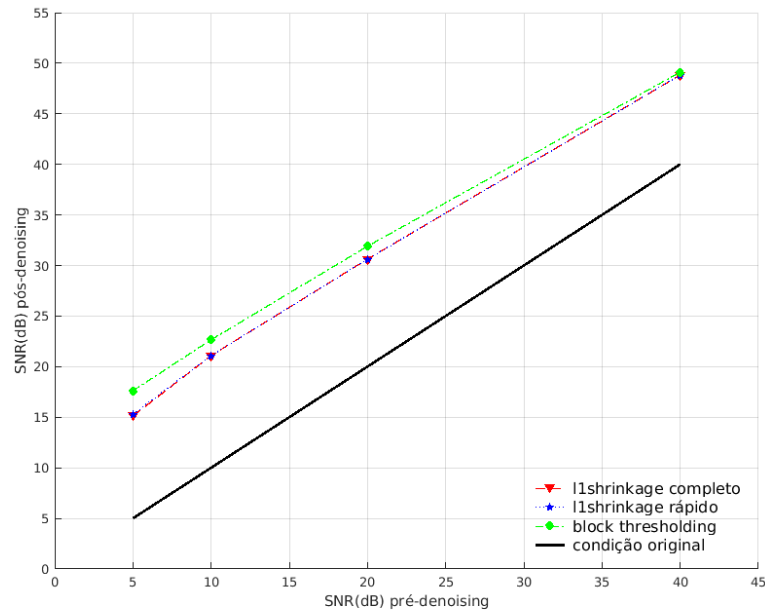
Tabela 9 – SNR média (dB) do sinal ECG.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	10,9 ± 0,4	11,2 ± 0,3	12,7 ± 0,3
10,0	16,7 ± 0,2	16,6 ± 0,2	17,8 ± 0,2
20,0	26,0 ± 0,2	25,7 ± 0,2	26,6 ± 0,2
40,0	43,6 ± 0,3	42,5 ± 0,3	42,0 ± 0,1

Conforme exposto na Figura 28, o ganho na SNR para o sinal ECG foi menor que nos demais sinais, o que se dá pela baixa taxa de amostragem, 256Hz. É possível observar, novamente, que o *denoising* por bloco tempo-frequência tende a apresentar maiores ganhos na SNR para situações iniciais piores.

Para avaliar o efeito da taxa de amostragem nas três técnicas, o teste com o sinal ECG foi realizado novamente para uma amostragem de 1024Hz. A Figura 29 e a Tabela 10 apresentam a SNR do sinal recuperado em função da SNR inicial para o sinal ECG amostrado a 1024Hz.

Figura 29 – Desempenho médio para o sinal ECG a 1024Hz.



Fonte: O próprio autor

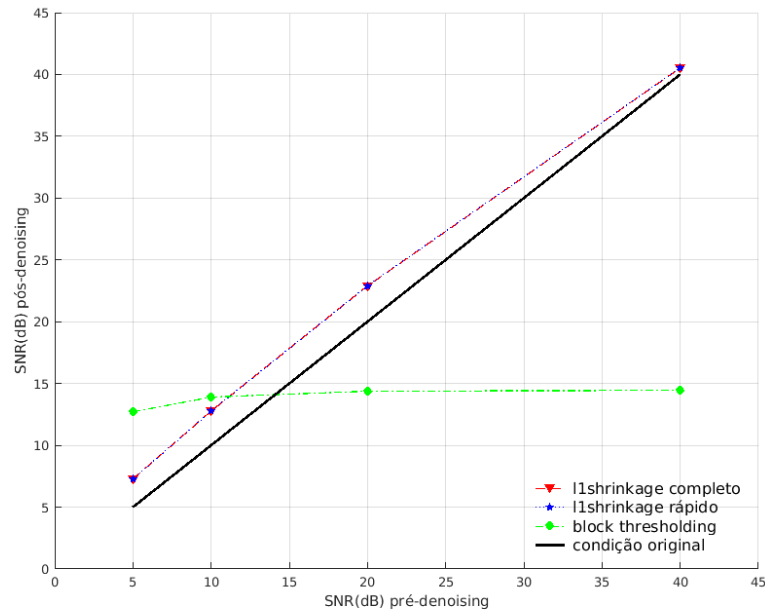
Tabela 10 – SNR média (dB) do sinal ECG a 1024Hz.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	15,1 ± 0,4	15,3 ± 0,2	17,6 ± 0,3
10,0	21,1 ± 0,2	21,1 ± 0,2	22,7 ± 0,2
20,0	30,6 ± 0,2	30,6 ± 0,2	31,9 ± 0,1
40,0	48,8 ± 0,2	48,8 ± 0,2	49,1 ± 0,1

Na Figura 29, é possível observar que os ganhos na SNR do sinal ECG a 1024Hz foram maiores que no caso anterior (Figura 28) em todas as técnicas, em especial para o *denoising* por bloco tempo-frequência. Em contrapartida, o tempo de processamento é elevado de forma significativa, como será observado na Seção 4.2.

A Figura 30 e a Tabela 11 apresentam a SNR do sinal recuperado em função da SNR inicial para o sinal Mozart.

Figura 30 – Desempenho médio para o sinal Mozart.



Fonte: O próprio autor

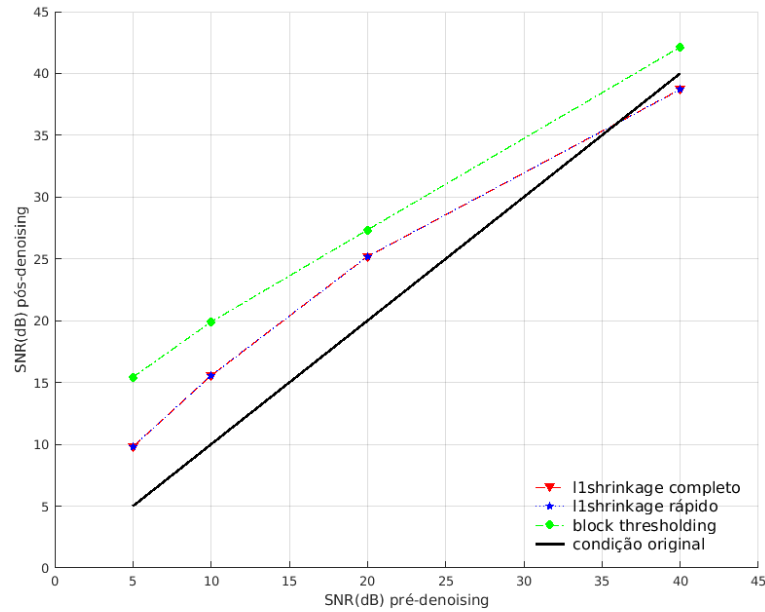
Tabela 11 – SNR média (dB) do sinal Mozart.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	$7,3 \pm 0,01$	$7,3 \pm 0,02$	$12,7 \pm 0,06$
10,0	$12,8 \pm 0,01$	$12,8 \pm 0,01$	$13,9 \pm 0,03$
20,0	$22,9 \pm 0,01$	$22,9 \pm 0,02$	$14,4 \pm 0,02$
40,0	$40,5 \pm 0,02$	$40,5 \pm 0,03$	$14,4 \pm 0,01$

Para o sinal Mozart, observou-se (Figura 30) que o método por bloco tempo-frequência diminuiu a SNR do sinal para SNR's maiores que $20dB$. Nesses casos, devido a complexidade do sinal, o método de Garcia (2010) sobrestimou de forma significativa a potência do ruído, o que comprometeu o desempenho do *denoiser*. Observou-se, também, que o ganho na SNR pelos métodos baseados na DWT foi pequena, em relação aos demais sinais. Isto se dá pela máxima escala de busca fixada em 1, menor que nos demais casos.

Para observar a influência da estimativa do ruído no *block thresholding*, assim como a escala máxima de busca no *l1shrinkage*, o experimento foi realizado novamente com parâmetros alterados. Para o *block thresholding*, utilizou-se o desvio padrão real do ruído, enquanto a escala máxima de busca para o *l1shrinkage* foi fixada em 2. A SNR média deste experimento está apresentada na Figura 31 e na Tabela 12.

Figura 31 – Desempenho médio para o sinal Mozart com desvio padrão real do ruído e escala máxima de 2.



Fonte: O próprio autor

Tabela 12 – SNR média (dB) do sinal Mozart com desvio padrão real do ruído e escala máxima de 2.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	9,8 ± 0,02	9,8 ± 0,02	15,4 ± 0,06
10,0	15,6 ± 0,02	15,6 ± 0,02	19,9 ± 0,05
20,0	25,2 ± 0,11	25,2 ± 0,11	27,3 ± 0,03
40,0	38,7 ± 0,08	38,7 ± 0,08	42,1 ± 0,01

Conforme observado na Figura 31, o desempenho do *block thresholding* foi superior ao teste anterior (Figura 30), o que demonstra sua dependência de uma boa estimativa da potência do ruído. Assim, o método tem grande potencial de recuperação, porém a necessidade desta estimativa como parâmetro de entrada do *denoiser* deve ser considerado como uma limitação, visto que aumenta o custo computacional e potencialmente diminui a eficiência de recuperação do sinal.

Para os dois métodos baseados na DWT, o ganho na SNR foi, em média, 2.5dB maior que no caso anterior para SNR's iniciais menores que 20dB. Porém, para a SNR inicial igual a 40dB, o *denoiser* produziu um sinal pior que a condição inicial, o que não aconteceu para escala máxima de busca fixada em 1. Especula-se, então, que escalas maiores devem ser utilizadas para sinais com baixa SNR.

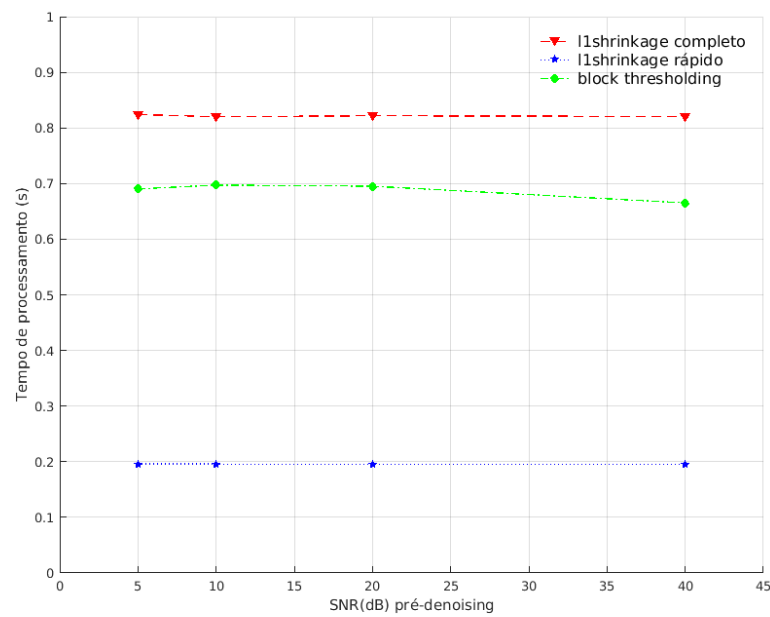
A partir do desempenho apresentado pelos métodos *l1shrinkage* completo e rápido, exposto nas Tabelas 5, 6, 7, 8, 9, 10, 11 e 12, o teste de Wilcoxon foi aplicado e resultou

em $p = 94,9\%$. Portanto, é plausível concluir que não houve diferença significativa entre o desempenho apresentado pelos dois métodos.

4.2 Tempo de Processamento

A Figura 32 e a Tabela 13 apresentam o tempo de processamento, em ms, em função da SNR inicial para o sinal *blocks*.

Figura 32 – Tempo de processamento médio para o sinal *blocks*.



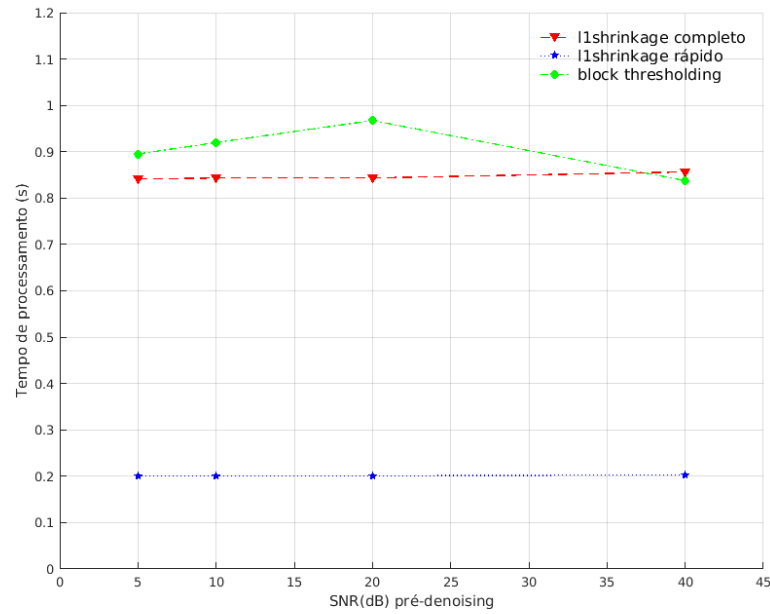
Fonte: O próprio autor

Tabela 13 – Tempo de processamento médio (ms) do sinal *blocks*.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	824,0 ± 10,2	195,8 ± 3,5	690,2 ± 13,2
10,0	820,2 ± 9,1	195,8 ± 2,2	697,2 ± 13,4
20,0	821,8 ± 10,2	195,2 ± 2,6	695,2 ± 13,2
40,0	820,6 ± 9,7	195,4 ± 3,4	665,2 ± 12,7

Na Figura 32, é possível observar que o tempo de processamento das três técnicas não depende significativamente da SNR inicial do sinal. Para todos os casos, o *l1shrinkage* rápido apresentou o menor tempo de processamento.

A Figura 33 e a Tabela 14 apresentam o tempo de processamento, em ms, em função da SNR inicial para o sinal *bumps*.

Figura 33 – Tempo de processamento médio para o sinal *bumps*.

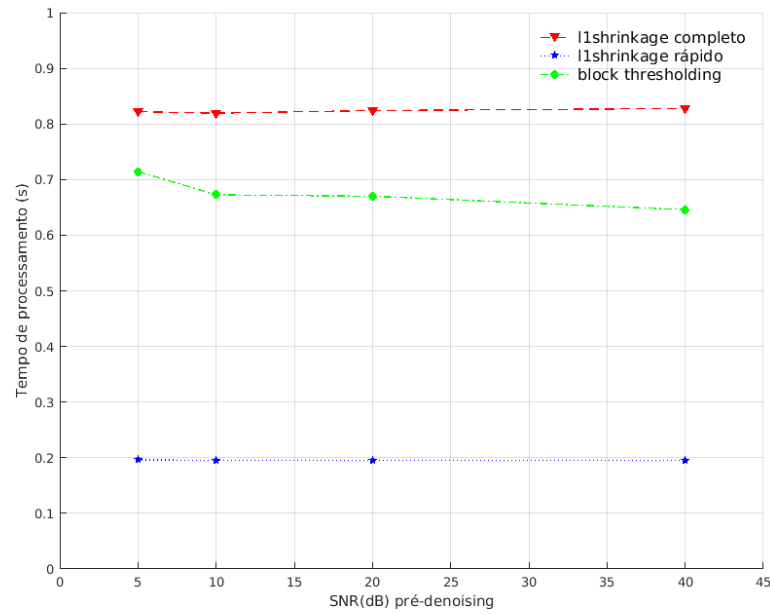
Fonte: O próprio autor

Tabela 14 – Tempo de processamento médio (ms) do sinal *bumps*.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	841,0 ± 8,0	200,6 ± 2,8	895,3 ± 165,4
10,0	843,3 ± 7,4	200,8 ± 1,6	919,7 ± 162,9
20,0	843,4 ± 10,4	200,9 ± 1,9	967,3 ± 149,0
40,0	856,3 ± 11,0	202,9 ± 2,2	937,6 ± 119,4

Novamente, é possível observar (Figura 33) que o *l1shrinkage* rápido apresentou o menor tempo de processamento dentre as técnicas testadas.

A Figura 34 e a Tabela 15 apresentam o tempo de processamento, em ms, em função da SNR inicial para o sinal *doppler*.

Figura 34 – Tempo de processamento médio para o sinal *doppler*.

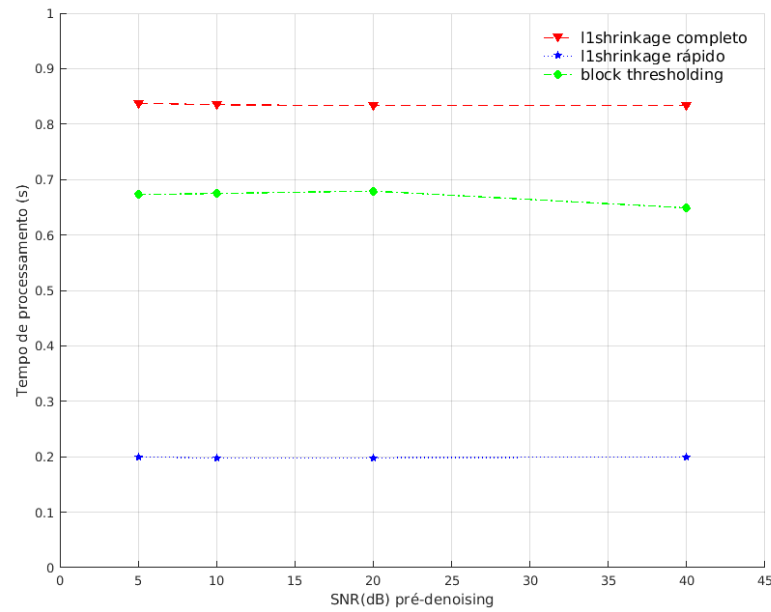
Fonte: O próprio autor

Tabela 15 – Tempo de processamento médio (ms) do sinal *doppler*.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	822,1 ± 10,1	196,2 ± 3,7	714,4 ± 22,1
10,0	819,3 ± 6,9	195,5 ± 2,4	672,5 ± 13,2
20,0	823,9 ± 12,6	195,3 ± 1,4	670,0 ± 10,8
40,0	827,6 ± 7,8	195,3 ± 1,8	645,8 ± 9,0

Novamente, é possível observar (Figura 34) que o tempo de processamento não depende significativamente da SNR inicial do sinal. O *l1shrinkage* rápido apresentou o menor tempo de processamento dentre as técnicas testadas.

A Figura 35 e a Tabela 16 apresentam o tempo de processamento, em ms, em função da SNR inicial para o sinal *heavy sine*.

Figura 35 – Tempo de processamento médio para o sinal *heavy sine*.

Fonte: O próprio autor

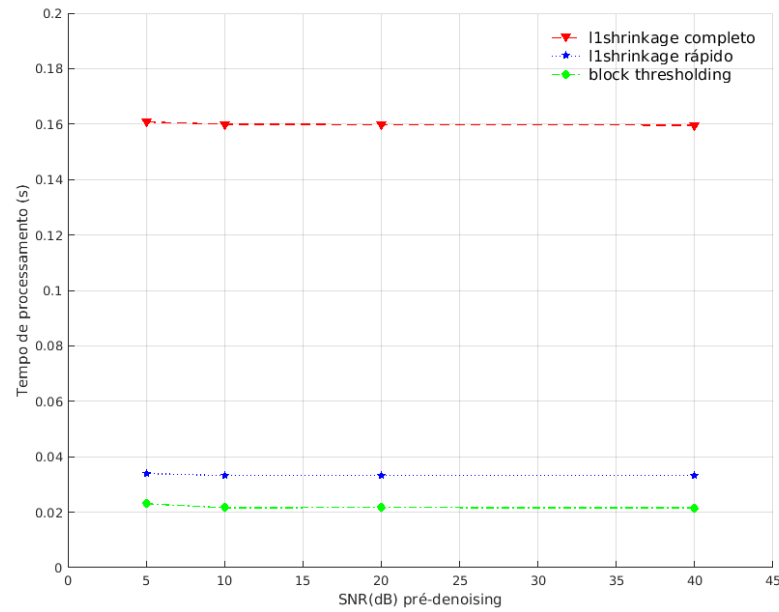
Tabela 16 – Tempo de processamento médio (ms) do sinal *heavy sine*.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	837,6 ± 8,8	200,0 ± 5,3	672,8 ± 9,0
10,0	834,8 ± 6,7	197,5 ± 1,9	674,8 ± 12,2
20,0	833,2 ± 3,2	198,1 ± 1,5	679,2 ± 9,8
40,0	833,5 ± 2,3	199,7 ± 6,6	648,8 ± 8,9

Observou-se (Figuras 32, 33, 34 e 35) que o tempo de processamento dos *denoisers* não apresentou muita variação com o sinal aplicado, porém grande divergência em função da técnica utilizada. O *l1shrinkage* rápido apresentou o menor tempo de processamento nos quatro sinais, enquanto o *l1shrinkage* completo foi a técnica mais lenta na maioria dos casos.

A Figura 36 e a Tabela 17 apresentam o tempo de processamento, em ms, em função da SNR inicial para o sinal ECG.

Figura 36 – Tempo de processamento médio para o sinal ECG.



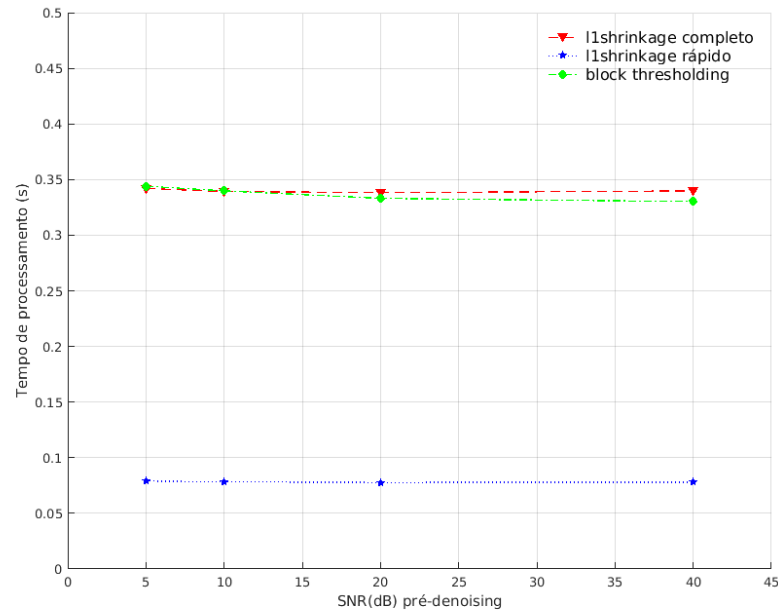
Fonte: O próprio autor

Tabela 17 – Tempo de processamento médio (ms) do sinal ECG.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block</i> <i>thresholding</i>
5,0	160,8 ± 1,5	33,9 ± 2,6	23,0 ± 2,8
10,0	159,9 ± 0,6	33,2 ± 0,5	21,6 ± 0,1
20,0	159,8 ± 0,6	33,2 ± 0,2	21,7 ± 0,1
40,0	159,6 ± 0,6	33,2 ± 0,3	21,6 ± 0,1

Devido a baixa taxa de amostragem (256Hz), o tempo de processamento para o sinal ECG (Figura 36) for menor que nos demais sinais em todas as técnicas, em especial no *block thresholding*, que apresentou o menor tempo em todos os casos. Para o sinal ECG amostrado a 1024Hz, a Figura 37 e a Tabela 18 apresentam o tempo de processamento, em ms, em função da SNR inicial.

Figura 37 – Tempo de processamento médio para o sinal ECG a 1024Hz.



Fonte: O próprio autor

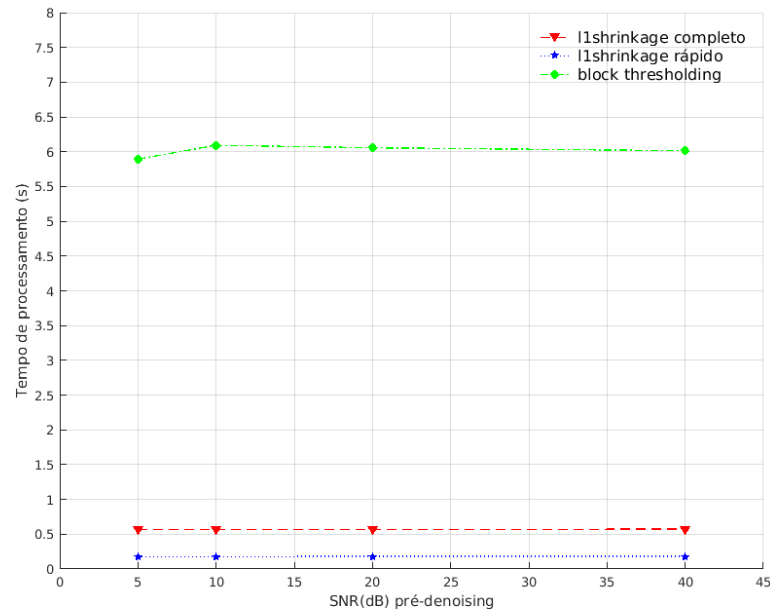
Tabela 18 – Tempo de processamento médio (ms) do sinal ECG a 1024Hz.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block</i> <i>thresholding</i>
5,0	341,9 ± 4,8	79,0 ± 2,6	344,0 ± 11,3
10,0	339,3 ± 3,1	78,3 ± 1,0	339,8 ± 8,1
20,0	338,2 ± 3,9	77,7 ± 0,5	333,1 ± 8,8
40,0	339,8 ± 1,9	78,1 ± 0,7	330,2 ± 7,5

Na Figura 37, é possível observar que para a maior taxa de amostragem, houve um aumento no tempo de processamento para as três técnicas, porém o aumento mais significativo foi na *block thresholding*.

A Figura 38 e a Tabela 19 apresentam o tempo de processamento, em ms, em função da SNR inicial para o sinal Mozart.

Figura 38 – Tempo de processamento médio para o sinal Mozart.



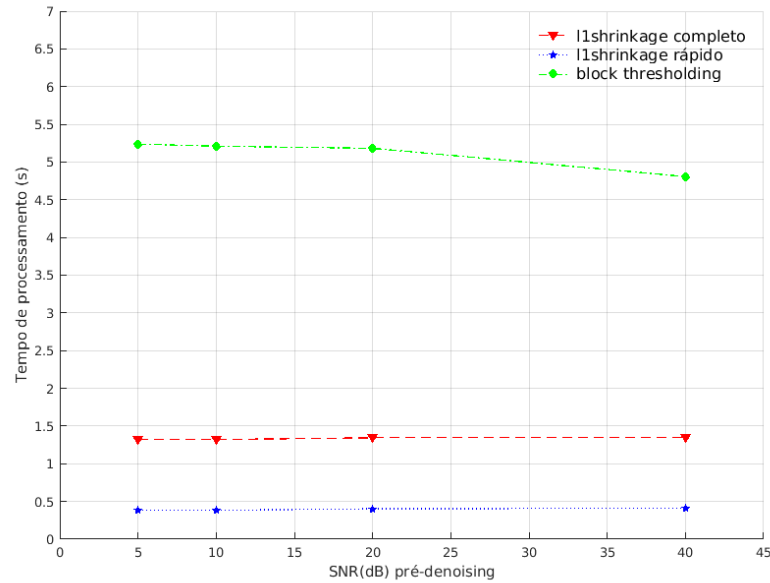
Fonte: O próprio autor

Tabela 19 – Tempo de processamento médio (ms) do sinal Mozart.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	562,3 ± 7,9	178,0 ± 6,2	5894,2 ± 183,8
10,0	565,5 ± 9,0	177,6 ± 4,7	6092,8 ± 105,2
20,0	568,9 ± 6,9	181,3 ± 6,0	6056,6 ± 133,7
40,0	572,4 ± 6,6	179,7 ± 1,1	6014,3 ± 128,1

Para o sinal Mozart, devido a elevada amostragem de 11kHz, a diferença no tempo de processamento (Figura 38) entre as técnicas foi a maior dentre os casos analisados. A *block thresholding* foi, em média, 33 vezes mais lenta que o *l1shrinkage* rápido, que foi apenas três vezes mais rápido que o método completo. Para o segundo teste realizado com o sinal Mozart a Figura 39 e a Tabela 20 apresentam o tempo de processamento, em ms, em função da SNR inicial,

Figura 39 – Tempo de processamento médio para o sinal Mozart com desvio padrão real do ruído e escala máxima de 2.



Fonte: O próprio autor

Tabela 20 – Tempo de processamento médio (ms) do sinal Mozart com desvio padrão real do ruído e escala máxima de 2.

SNR inicial	<i>l1shrinkage</i> completo	<i>l1shrinkage</i> rápido	<i>block thresholding</i>
5,0	1319,4 ± 11,0	383,3 ± 3,3	5238,8 ± 51,7
10,0	1321,2 ± 8,8	382,8 ± 2,9	5209,3 ± 45,3
20,0	1344,4 ± 16,0	402,8 ± 9,1	5182,6 ± 32,6
40,0	1345,9 ± 7,8	409,6 ± 6,7	4807,2 ± 16,6

Devido o aumento na escala máxima de busca em relação ao teste anterior, o tempo de processamento (Figura 39) nas técnicas baseadas na transformada *wavelet* aumentou, em média, duas vezes, enquanto o tempo gasto pelo *denoiser* por bloco tempo-frequência diminuiu. Porém, este manteve-se com tempo superior as demais técnicas.

5 Conclusões e Trabalhos Futuros

Este trabalho apresentou duas metodologias, denominadas *l1shrinkage* completo e *l1shrinkage* rápido, para a adaptação dos parâmetros de um algoritmo de *denoising* por transformada *wavelet*. Ambas estão baseadas na relação entre a esparsidade da representação de um sinal e a eficiência do *denoiser* que, como constatou-se, não é estritamente decrescente, apesar da tendência global de que, dentre um conjunto de representações, a representação mais esparsa resulte no maior desempenho de recuperação. Os métodos propostos são resultados de dois modos distintos de buscar a base que produz a representação mais esparsa: a busca exaustiva (*l1shrinkage* completo) e busca em dicionário reduzido (*l1shrinkage* rápido).

Para os sinais *blocks*, *bumps*, *doppler* e *heavy sine*, o *l1shrinkage* completo e o *l1shrinkage* rápido apresentaram ganho na SNR do sinal pouco variante com o sinal aplicado e SNR inicial. Isto demonstra a abrangência das técnicas, especialmente em contraposição com o método de Yu, Mallat e Bacry (2008), visto que mesmo em sinais bem diferentes, como *blocks* e *doppler*, o método produziu resultados satisfatórios. A diferença de desempenho entre os *denoisers* por transformada *wavelet* não foi significativa na maior parte dos casos. No sinal de eletrocardiografia, o *denoising* por *block thresholding* apresentou, em média, desempenho superior às demais técnicas, especialmente com taxa de amostragem de 1024Hz. No sinal de áudio (Mozart), o *block thresholding* apenas produziu SNR's maiores que as técnicas propostas para SNR's iniciais inferiores a 20dB, devido a sua grande dependência da estimação da potência do ruído.

Um teste de hipótese foi aplicado para a verificação da diferença de desempenho entre os métodos *l1shrinkage* completo e rápido. Com $p = 94,9\%$ resultante, é plausível aceitar a hipótese nula de que, apesar do reduzido dicionário e custo computacional, o método rápido não apresenta diferença significativa de desempenho em relação ao método completo.

Quanto ao tempo de processamento, o *l1shrinkage* rápido apresentou a menor média em todos os sinais testados, com exceção do sinal de ECG a 256Hz. Foi observado que a diferença de tempo de processamento entre as técnicas baseadas na DWT e o *block thresholding* é acentuada com o aumento da taxa de amostragem.

Ambas as técnicas propostas apresentam um único parâmetro de entrada, a escala máxima de busca. Portanto, são significativamente menos dependentes do usuário que o *denoising* por *wavelet shrinkage* original e por *block thresholding*. Este último depende da largura temporal da janela de Hanning e da estimação do desvio padrão do ruído, que influenciam largamente o desempenho.

5.1 Trabalhos Futuros

Como decorrência do presente trabalho, destacam-se dois temas a serem estudados em trabalhos futuros.

Primeiramente, os métodos aqui apresentados limitaram-se a transformações ortogonais. Isto se justifica pelo menor custo computacional tanto na síntese do sinal quanto na verificação da representação mais esparsa. Como proposta para trabalhos futuros, portanto, o estudo da utilização de transformações não ortogonais pode ser realizado para verificar se, apesar do acréscimo no custo computacional, o desempenho pode ser elevado.

Em segundo lugar, ambas as técnicas propostas ainda dependem de um parâmetro de entrada, a máxima escala de busca. Foi observado que este parâmetro está relacionado à complexidade do sinal, taxa de amostragem e SNR inicial. Propõe-se, portanto, que esta relação seja mais profundamente estudada, em especial no vetor de aproximação. Com esta relação bem estabelecida, especula-se que a escala máxima de busca possa ser definida a partir do próprio sinal, o que tornaria o método inteiramente adaptativo.

Referências

- AHARON, M. et al. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, IEEE, v. 54, n. 11, p. 4311, 2006. Citado na página 14.
- BIGOT, J.; DELEDALLE, C.; FÉRAL, D. Generalized sure for optimal shrinkage of singular values in low-rank matrix denoising. *Journal of Machine Learning Research*, v. 18, n. 137, p. 1–50, 2017. Citado na página 14.
- CHANG, S. G.; YU, B.; VETTERLI, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE transactions on image processing*, IEEE, v. 9, n. 9, p. 1532–1546, 2000. Citado na página 14.
- DAUBECHIES, I. *Ten lectures on wavelets*. [S.l.]: Siam, 1992. v. 61. Citado 3 vezes nas páginas 18, 23 e 30.
- DONOHOO, D. L.; JOHNSTONE, I. M. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, Taylor & Francis Group, v. 90, n. 432, p. 1200–1224, 1995. Citado 6 vezes nas páginas 14, 17, 21, 26, 27 e 33.
- DONOHOO, D. L.; JOHNSTONE, J. M. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, Oxford University Press, v. 81, n. 3, p. 425–455, 1994. Citado 4 vezes nas páginas 14, 17, 25 e 26.
- GARCIA, D. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational statistics & data analysis*, Elsevier, v. 54, n. 4, p. 1167–1178, 2010. Citado 2 vezes nas páginas 39 e 46.
- GIBBONS, J. D.; CHAKRABORTI, S. Nonparametric statistical inference. In: *International encyclopedia of statistical science*. [S.l.]: Springer, 2011. p. 977–979. Citado na página 64.
- HE, Y. et al. Adaptive denoising by singular value decomposition. *IEEE Signal Processing Letters*, IEEE, v. 18, n. 4, p. 215–218, 2011. Citado na página 14.
- HOYER, P. O. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, v. 5, n. Nov, p. 1457–1469, 2004. Citado na página 32.
- HURLEY, N.; RICKARD, S. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, IEEE, v. 55, n. 10, p. 4723–4741, 2009. Citado 2 vezes nas páginas 31 e 32.
- HYVÄRINEN, A.; HOYER, P. O.; OJA, E. Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 1999. p. 473–479. Citado na página 14.
- KRIM, H. et al. On denoising and best signal representation. *IEEE transactions on information theory*, IEEE, v. 45, n. 7, p. 2225–2238, 1999. Citado na página 13.

- LUISIER, F.; BLU, T.; UNSER, M. A new sure approach to image denoising: Interscale orthonormal wavelet thresholding. *IEEE Transactions on image processing*, IEEE, v. 16, n. 3, p. 593–606, 2007. Citado na página 14.
- MALLAT, S. *A wavelet tour of signal processing the Sparse way*. 3. ed. [S.l.]: Academic Press, 2008. Citado 3 vezes nas páginas 14, 18 e 22.
- MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, Ieee, v. 11, n. 7, p. 674–693, 1989. Citado 4 vezes nas páginas 18, 19, 20 e 21.
- MCSHARRY, P. E. et al. A dynamical model for generating synthetic electrocardiogram signals. *IEEE transactions on biomedical engineering*, IEEE, v. 50, n. 3, p. 289–294, 2003. Citado na página 38.
- METZLER, C. A.; MALEKI, A.; BARANIUK, R. G. From denoising to compressed sensing. *IEEE Transactions on Information Theory*, IEEE, v. 62, n. 9, p. 5117–5144, 2016. Citado 2 vezes nas páginas 13 e 14.
- SARDY, S. Minimax threshold for denoising complex signals with waveshrink. *IEEE Transactions on Signal Processing*, IEEE, v. 48, n. 4, p. 1023–1028, 2000. Citado na página 14.
- STARCK, J.-L.; CANDÈS, E. J.; DONOHO, D. L. The curvelet transform for image denoising. *IEEE Transactions on image processing*, IEEE, v. 11, n. 6, p. 670–684, 2002. Citado na página 14.
- STEIN, C. M. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, JSTOR, p. 1135–1151, 1981. Citado na página 25.
- YU, G.; MALLAT, S.; BACRY, E. Audio denoising by time-frequency block thresholding. *IEEE Transactions on Signal processing*, IEEE, v. 56, n. 5, p. 1830–1839, 2008. Citado 5 vezes nas páginas 14, 16, 38, 39 e 56.
- ZHANG, X.-P.; DESAI, M. D. Adaptive denoising based on sure risk. *IEEE signal processing letters*, IEEE, v. 5, n. 10, p. 265–267, 1998. Citado na página 14.

APÊNDICE A – Pseudo-códigos

Algoritmo 1: Transformada discreta multiresolução de *wavelet*

Entrada: Vetor x de tamanho N , filtros H e G e escala L .

$l \leftarrow 1$

$D \leftarrow []$

enquanto $l \leq L$ **faça**

$d \leftarrow x * G$

$x \leftarrow x * H$

$x' \leftarrow d' \leftarrow []$

para n *de* 1 *a* N *passo* 2 **faça**

$x' \leftarrow [x' \quad x(n)]$

$d' \leftarrow [d' \quad d(n)]$

fim

$x \leftarrow x'$

$D \leftarrow [D \quad d']$

$N \leftarrow N/2$

$l \leftarrow l + 1$

fim

Saída: Vetor $w = [x \quad D]$.

Algoritmo 2: Limiar que minimiza o risco de Stein

Entrada: Vetor x de tamanho N .

$x_2 \leftarrow x^2$

Ordenar x_2

$R \leftarrow []$

$x_{2sum} \leftarrow 0$

para n *de* 1 *a* N **faça**

$x_{2sum} \leftarrow x_{2sum} + x_2(n)$

$r \leftarrow \frac{N-2n+x_{2sum}+(N-n).x_2(n)}{N}$

$R \leftarrow [R \quad r]$

fim

Saída: Limiar $t_s = \sqrt{x_2(i)}$, sendo i o índice do menor elemento do vetor R .

Algoritmo 3: *Denoising por wavelet shrinkage***Entrada:** Vetor x de tamanho N , filtros H e G e escala L . $w \leftarrow DWT$ na escala L do vetor x $t_s \leftarrow$ limiar que minimiza $SURE$ do vetor w $w' \leftarrow []$ **para** n de 1 a N **faça** **se** $|w(n)| \geq t_s$ **então** $w' \leftarrow [w' \quad \text{sgn}(w(n)).(|w(n)| - t_s)]$ **senão** $w' \leftarrow [w' \quad 0]$ **fim****fim****Saída:** Vetor \hat{x} , resultante da $IDWT$ do vetor w' .**Algoritmo 4:** *Denoising por $l1$ shrinkage completo***Entrada:** Vetor x e escala máxima de busca L . $wavelets = \{'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10', 'coif1', 'coif2', 'coif3', 'coif4', 'coif5', 'dmey'\}$ **para** i de 1 a 23 **faça** **para** l de 1 a L **faça** $w \leftarrow DWT$ na base $wavelets[i]$, escala l do vetor x **se** $i = 1$ e $l = 1$ **então** $[w_{best} \quad l_{best} \quad l1_{min}] \leftarrow [i \quad l \quad |w|_1]$ **senão** **se** $|w|_1 < l1_{min}$ **então** $[w_{best} \quad l_{best} \quad l1_{min}] \leftarrow [i \quad l \quad |w|_1]$ **fim** **fim****fim****fim****Saída:** Vetor \hat{x} , resultante do *wavelet shrinkage* do vetor x na base w_{best} e escala l_{best} .

Algoritmo 5: *Denoising* por *l1 shrinkage* rápido

Entrada: Vetor \mathbf{x} e escala máxima de busca L . $wavelets = \{'db1', 'db10', 'sym4', 'coif5', 'dmey'\}$ **para** i de 1 a 23 **faça** **para** l de 1 a L **faça** $w \leftarrow DWT$ na base $wavelets[i]$, escala l do vetor x **se** $i = 1$ e $l = 1$ **então** $[w_{best} \quad l_{best} \quad l1_{min}] \leftarrow [i \quad l \quad |w|_1]$ **senão** **se** $|w|_1 < l1_{min}$ **então** $[w_{best} \quad l_{best} \quad l1_{min}] \leftarrow [i \quad l \quad |w|_1]$ **fim** **fim** **fim****fim****Saída:** Vetor $\hat{\mathbf{x}}$, resultante do *wavelet shrinkage* do vetor x na base w_{best} e escala l_{best} .

APÊNDICE B – Teste Estatístico de Wilcoxon

O teste estatístico de Wilcoxon é um teste não paramétrico que avalia um conjunto de observações aleatórias a partir da hipótese nula de que sua mediana é igual a um valor M_o . Quando utilizado na comparação de desempenho entre dois métodos distintos, como neste trabalho, as observações aleatórias consistem na diferença entre os desempenhos dos métodos e a mediana M_o é fixada em 0. Como resultado do teste, um valor $p \leq 1$ indica a confiança na aceitação ou rejeição da hipótese nula. Normalmente, se $p < 5\%$ a hipótese nula é rejeitada (GIBBONS; CHAKRABORTI, 2011).

Matematicamente, dados dois vetores X e Y , o vetor de diferenças não nulas $D[n] = X[n] - Y[n]$ de tamanho N é ordenado pela magnitude de seus componentes. Para cada elemento i de D , então, é atribuído um posto R_i de 1 a N de acordo com a ordem crescente das magnitudes de $D[i]$. Em casos de empate, a média dos postos é atribuída. Define-se, portanto (GIBBONS; CHAKRABORTI, 2011):

$$R^+ = \sum_{D[i]>0} R_i, \quad (\text{B.1})$$

e

$$R^- = \sum_{D[i]<0} R_i. \quad (\text{B.2})$$

Sendo o valor W tal que $W = \min(R^+, R^-)$ e R_E o número de empates na atribuição de postos ao vetor D , a estatística:

$$Z = \frac{W - \frac{N(N+1)}{4}}{\frac{N(N+1)(2N+1)}{24} - R_E}, \quad (\text{B.3})$$

é aproximadamente normal para $N > 25$. Assim, sendo $F : \mathbb{R} \rightarrow \mathbb{R}^+$ a função de distribuição cumulativa (cdf) normal, o valor p é dado por:

$$p = 2.F(z|0, 1). \quad (\text{B.4})$$